Software Design

软智慧 硬力量

OS Network IT环境 **综合技术**

02

中文版

（日）技术评论社 编

ソフトウェア デザイン

Makes New Vision ·························· 从理论到实践

3D打印机：拓展未来的理由

3D打印的原理 · 安装 · 操作 · 选购

以系统洞察力驾驭软件开发

软件评测中的BUG排查三大技巧

分布式数据库“未来工房”

何为高可用性——Riak功能探究与测试



安全实践鬼手诀

目标式攻击和APT攻击

可以使用SDN吗？应该使用SDN吗

网络虚拟化的陷阱之OpenFlow

Android工程师的邀请函

在FPGA上搭载Android

人民邮电出版社
POSTS & TELECOM PRESS

Software Design 中文版02

（日）技术评论社 编

3D打印 + Debug技术

Software Design 02

中文版

人民邮电出版社
北京

数字版权声明

图灵社区的电子书没有采用专有客户端，您可以在任意设备上，用自己喜欢的浏览器和PDF阅读器进行阅读。

但您购买的电子书仅供您个人使用，未经授权，不得进行传播。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

图书在版编目（C I P）数据

Software Design中文版. 2 / 技术评论社编. -- 北京 : 人民邮电出版社, 2014.7
ISBN 978-7-115-35812-7

Ⅰ. ①S… Ⅱ. ①技… Ⅲ. ①软件开发 Ⅳ. ①TP311.52

中国版本图书馆CIP数据核字(2014)第112635号

内 容 提 要

*Software Design*是日本主流的计算机技术杂志，旨在帮助程序员更实时、深入地了解前沿技术，扩大视野，提升技能。内容侧重于网络、操作系统、开源软件和信息处理技术等。

本期的主题为：3D打印和Debug技术。特辑1详细讲解了3D打印当前的现状、3D打印机的原理和构造，并教读者动手进行3D建模并打印输出，最后对如何选择3D打印机给出了宝贵建议。特辑2将社会统计学等方法用于软件评测，总结出数个实用的Debug方法，有利于开发者提高开发速度与质量。本书适合各行业软件开发者阅读。

◆ 编　　　[日]技术评论社
责任编辑　乐　馨
执行编辑　武晓宇　高宇涵
责任印制　焦志炜
装帧设计　broussaille私制

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京　　　　　　　　印刷

◆ 开本：787×1092　1/16
印张：10.25
字数：258千字　　　　2014年7月第1版
印数：1-3 500册　　　2014年7月北京第1次印刷
著作权合同登记号　图字：01-2014-0496号

定价：39.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第0021号

目录

Software Design 中文版 02

Contents

分栏目录

Software Design 中文版 02

Contents By Topics

特辑1

从理论到实践

3D打印机拓展未来的理由

特辑2

狙击Bug的技术

以系统洞察力(眼力)驾驭软件开发

专　栏

软件开发

操作系统/网络

专家视点

2

第56回

IT工程师必须知道的 最新术语

文/杉山贵章 Sugiyama Takaaki（ONGS公司）
takaaki@ongs.co.jp
译/苏祎

次时代的Web浏览器内核引擎

动荡的Web浏览器市场

Web浏览器所搭载的浏览器内核，是支撑以HTML5/CSS3为中心的Web技术快速发展的最重要因素。近年，Web浏览器内核势力鼎足三分，主要有Apple公司和Google公司开发的、Safari和Chrome（以及Chromium）所搭载的WebKit；Mozilla Foundation开发的、Firefox所搭载的Gecko；还有Microsoft公司所开发、Internet Explorer搭载的Trident。

特别是WebKit，在2005年开源后，它的份额急速扩大，现在基于WebKit内核的Web浏览器已经占领了大约一半的市场。其快速成长的背后，一方面源于与最新Web技术相对应的、技术与硬件方面的高强度发展；另一方面，源于移动终端市场中，iOS（Safari被用作默认浏览器）和Android（Chrome被用作默认浏览器）拥有压倒性的份额。

2013年2月，Opera公司也发表声明，将把Web浏览器Opera的内核引擎从自己公司研发的Presto更换为WebKit，WebKit的市场份额更加坚实。但是，进入4月，传出了两条冲击性的新闻，撼动着原有的势力版图。第一条是，Mozilla宣布，将与三星联手开发替换Gecko的最新引擎Servo。第二条是，Google宣布，将开发用于Chrome的新引擎Blink，用以替换WebKit。

Google的新引擎Blink

Blink以WebKit项目的分支形式进行开发，最大的特征是，基础代码的完善与多进程架构的优化。Blink将从Chrome 28开始正式使用，很快会发布正式版（截至本文编写时）。但是，因其涉及内部的架构改动与代码的简化，这是否会对Web开发者产生较大的影响仍不明了。虽然如此，刚刚决定使用WebKit内核的Opera仍将意愿移向天平中Blink的一端。

Google舍弃WebKit，进而独自开发内核引擎，主要是因为Chrome与其他的Web浏览器相比，采用了多进程架构。为了对应Chrome，WebKit的开发变得很繁琐，反过来，Chrome方面为了保持和WebKit的兼容性，也阻碍了它技术的进一步改良，两者最终在痛苦中做出了艰难的分手决定。

Mozilla的新引擎Servo

Servo是基于Rust语言开发的新引擎，主要功能是支持多核和异构处理器，灵活使用GPU进行并行处理，通过大幅改善内存管理来强化安全性。Rust是Mozilla独自推进开发的开源语言，主要特点是精准的硬件资源调配和内存安全功能，以及针对多核CPU的优化设计，因此，Rust具备Servo所要求的全部要素。

Servo最初是以Android+ARM这样的移动环境为目标进行开发的，并没有打算用于替换Gecko。Servo所扮演的角色是“针对新硬件的全新引擎”，PC环境下，依旧由Gecko担任主要的内核引擎。但是也不排除将来会把Servo的成果反馈到面向PC的浏览器中。

Blink和Servo这两个新引擎的出现，一方面扩大了Web浏览器的新的可能性，另一方面也包含了平台的多样性的问题。最先受其影响的Web开发者和设计师们，当然要密切关注各个公司的动向了。



网络工程师修炼手册

「自己家的服务器机架之推荐篇」

文/tomocha（http://tomocha.net/diary/）
译/阿逸
插画/高野凉香

第4回

机架的电源问题（2）



引　子

上期我们聊了电源的话题，本期我们将继续就电源进行讨论。

选择UPS

了解服务器机架的总用电量后，就可以研究如何选择UPS（不间断电源）了。UPS已经被视作必需品，但为什么一定要使用UPS？是为了提供不间断的电源供给吗？答案是否定的。UPS原本的作用是确保服务器等设备在断电后仍然有足够的时间安全关机，其他的功能还有保护用电设备免受电路间断、噪声、脉冲、电压不稳等的影响。UPS能够有效地保护设备免受供电问题的伤害，因此选用UPS是非常必要的。

UPS大致有3种供电方式：后备式（Offline）UPS、在线互动式UPS以及在线式UPS。

下面我将简单介绍下3种供电方式的差异。

● 后备式UPS

正常情况下，交流电源流经UPS直接给设备供电。在检测到停电的瞬间，UPS会切换到备用电池供电，转换时间一般在10 ms左右。后备式UPS结构简单、体型小巧、价格便宜，但是无法稳定输入电压。

● 在线互动式UPS

在线互动式UPS同后备式一样，正常情况下交流电源流经UPS直接给设备供电，但其内部有稳压器，可稳定电压、抗市电干扰。停电或者电压波动过大时，在线互动式UPS会切换到电池供电。同样，转换时间也为10 ms左右。此类UPS结构比较简单，成本较低，并且有稳定电压的作用。不过，若电压剧烈波动，电池会很快消耗完。

● 在线式UPS

该类UPS将输入的交流电转换为直流电，为电池充电的同时使用电池供电。因此可以始终保持稳定的电力输出。由于一直在进行直流电转换，所以用电量比起其他的UPS要大。

综上，在线式UPS性能最好，同时价格也是较高的，因此经常用于不间断（No Stop）系统以及对可用性要求较高的情况。若仅作为家用就有点显得杀鸡用牛刀了。如果不是非常贵重的设备，很少会使用在线式UPS。

面向家庭用户和SOHO的低价后备式UPS

不具备稳压、抗干扰功能，所以介于两者之间的在线互动式UPS就成为主流。各大生产商在售的主要产品就是在线互动式UPS，当然也有在线式UPS。

容量研究

确定UPS的种类后，我们再来研究UPS的容量。

UPS所需的容量，应该根据机架上设置并且使用设备的合计耗电量来确定。

UPS的容量以VA为单位表示。VA是伏特安培（Volt Ampere）的简称，为视在功率的单位。维基百科中将其定义如下：视在功率是正弦交流电电路中电压有效值与电流有效值的乘积。正如其字面意思，视在功率表示设备用电的容量。因此通过VA，可以事先得知实际用电量。有一些设备也会同时标明VA（视在功率）和W（功率）。

实际用电量可以通过下面的公式来计算。

VA × 功率因数 = W

（※ 一般服务器和网络设备的功率因数在0.7到0.8之间）

如果觉得上述计算公式麻烦，也可以这样：全部使用15 A供电回路的话，选择最大容量1500 VA的UPS就行了（实际消耗的电量小于1500 VA的话，后备供电的时间会变长）。但如果不留神买了3000 VA（3 KVA）的UPS，插座和插头需要符合L5-30（30 A用回路及插座）标准，还可能涉及到输电线路的施工，所以需要特别注意。家用插座的形状一般是像NEMA 5-15（图1）这样的。空调等使用20A回路的电器，多为NEMA5-20（图2）。虽然很多情况下都没有接地端子，但200 V的话就需要用NEMA6-XX/L6-XX（XX为安培数）了。

笔者个人比较偏好使用市面常见的APC Smart UPS机架安装型号。SUA1500RMJ2U（图3）属于在线互动式UPS，二手货1到2万日元（人民币约为560元到1200元）就能买到，电池方面性价比也比较合适，因此推荐这款。

最近的一些UPS，搭载了网络管理模块的可选功能，能够通过SNMP协议获取UPS的温度、负载等信息，还可以通过网络在备用电池运作的时候自动关闭服务器。请大家综合考虑需要的功能、容量、价格等因素，选择一款适合自己的UPS吧。

▼ 图1　NEMA 5-15插座的形状



▼ 图2　插座L5-20（左）以及NEMA 5-20（右）



▼ 图3　APC SmartUPS（SUA1500RMJ2U）



DIGITAL GADGET

新潮数码

文 / 安藤幸央　Yukio Ando（EXA公司）
Twitter：@yukio_andoh
个人主页：http://www.andoh.org/
译 / 苏祎

Volume 176 »»»» 让人爱不释手的童趣 Digital Gadget

玩儿转数码产品的“小鬼”们

最近经常可以看到小朋友们熟练使用 iPhone、iPad 等产品。大多数小朋友们会乖乖地看自己喜欢的卡通动画，或是集中精力玩涂鸦游戏等。他们生于互联网时代，与数码产品相熟相知，使用起来完全没有任何不适。据说还有一些小朋友们会触碰电视画面，并纳闷儿为何不能回放或检索放映中的电视节目。

虽说是小朋友，但也可以归类为 Kids 或是 Children。本文中提到的数码玩具，其设定的目标群体是相信圣诞老人存在的小朋友，年龄大约是3~12岁。

始于1996年的 Atomocom（http://www.atomo. com）是一个制作高互动性儿童玩具的老品牌。这家公司的 CEO，Bill McIntyre 先生在 UX Week 2012 的演讲中提到，小朋友们喜欢的智能手机应用、平板应用将具有以下特征：

- 将玩具融入到了画面中。
- 将画面巧妙组合，嵌入到玩具中。

数码玩具种类繁多，但最终能吸引住小朋友们的都是上述二者中的某一种。就是说，与没有实物感的完全数字化相比，以可以触摸、实际存在的“玩具”为媒介，能让小朋友获取更切实的感受。

从很久之前就存在一些玩乐方式，可以使小朋友的手部感觉更加灵敏，虽然看上去都是很随意的动作，但可以在无意识中锻炼手部的感觉。

- 撕碎纸的感觉。
- 将左右手中的东西互相敲打发出声音的感觉。
- 敲打物品的感觉，用手中握着的物品敲打物品发出声音的感觉。
- 以适当力度控制轻薄物体的能力，例如从盒子里抽出餐巾纸等。
- 夹起某些东西，将很小的、不稳定的东西夹到其他地方的动作。
- 拉扯某些东西。抓住有反弹力的东西。
- 模仿某些动作。
- 寻找被隐藏起来的东西。
- 通过手部、脸部动作等，感受



将玩具融入到了画面中的案例“Digital Barbie”



将画面嵌入到玩具中的应用案例“HappiTaps”



将画面嵌入到玩具中的应用案例“Woogie”

镜子中映射的内容。

- 享受物品的摇摆或旋转。
- 区别视觉与触觉下物体的差异。
- 维持物品、不稳定物品平衡感的感觉。把物品堆积起来的感觉。
- 眼睛与手的配合。明白可以通过眼睛去确认用手拨动的东西在动起来之后的情况。

另外，针对小朋友的应用或玩具，需要以不同于成人用户的角度来思考。大致内容如下：

- 小朋友们是否一眼就能看明白？
- 要细分年龄目标层。明确不同年龄段中小朋友的行为差异。
- 将男孩和女孩分开来考虑，或者是考虑男孩和女孩混在一起的情况。
- 要区分考虑多人使用的情况、小朋友独自使用的情况、和家长一起使用的情况。
- 若习惯了还能使用吗？是否可以通过学习形成习惯？是否会花费时间也无法习惯？
- 是否会有小朋友无法预想到的事情发生？

在考虑这些事项的基础上，再思考应用策划以及作为玩具的乐趣，但实际操作时还需要注意一些细节。所有人都曾经是小朋友，但大家都忘记了那时候的事情，更何况现在阅读这篇文章的读者，在还是小朋友时应该都没有平板设备，所以有很多事情是需要重新思考或学习的。针对小朋友的平板应用其设计要点和技巧如下：

- 小朋友的手指很小。虽然会有个体差异，但整体上手都很小。
- 小朋友在使用应用时，脸和屏幕间的距离往往会很近。
- 需要认识到，类似于“用户应该会这样使用吧”的前提或假设都是没有用的。
- 在有家长协助的前提下，要区分需要家长处理的部分（设置、购买等）与小朋友自行可以完成的部分。
- 设计上注重偶发性成功的体验，通过积累较小的成就感来提升总体乐趣。
- 注意维持技巧提升的乐趣，与集中力持续之间的平衡感（小朋友只能维持很短时间的注意力）。
- 要设置一些内容，让小朋友在经过一段时间后会想重新玩。
- 要考虑到小朋友会从任何奇怪的方向、用任意手指来触摸屏幕（不一定会是右手的食指）。
- 图画优于文字。即便是简单的文字、单词，也尽可能使用图来表示。但若是教育型产品，则不限于此。
- 小朋友比较擅长点击（Tap）或是滑动（swipe），但不擅长缩放或拖拽。
- 不可以用LOADING等进程条让他们等待。小朋友们一旦遇到需要等待的情况就不会使用了。

成人也可感受的童趣

比如乐高积木，质量好而且耐用，不仅适合小朋友，成人也可以拿它玩乐。再如乐器类玩具，使用非常轻松，通过练习就可以更加熟练地演奏，也属于小朋友和大人都可以享受的乐趣。

有一点应当注意，小朋友并不希望玩被人觉得有“孩子气”的玩具。大人们往往希望小朋友们玩“幼稚的、小朋友的”东西，但小朋友喜欢的却是“和大人一样”的东西。

从幼时开始熟悉数码设备是件非常好的事情。随着数码设备



将达芬奇的兵器手稿模型化的玩具（童友社）



人可以实际穿着的cosplay用的Iron Man服装。按订单生产，价格约180万日元（约11万人民币）（RooBeeze）



LEGO的星球大战人物。（LEGO）



LINE TOWN MY TOUCH。彩色液晶显示，可在近距离通信中交换贴图的儿童用设备。（TAKARA TOMY）

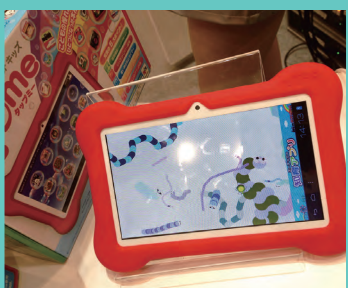
的发展，今后的生活中，不仅是教科书，教材的电子化，教育的形态也可能会发生变化。而且，对于小朋友来说，或许有很多感觉是只有在小时候可以感受或体验到的。

轻柔的、粗糙的、凹凸的、散乱的、黏黏的、干干的、皱皱的、不整齐的、黏稠的、刺刺的，等等。

这是用语言表达出的触觉感受。未来，这些感受可能会在数码设备中以数字化的方式呈现。但现在这种方式仍在研究中，现实社会中还未展开。

小朋友熟悉数码之前，还是希望他们能有绘画时颜料弄脏了手，或是手工制作中受到小伤之类的经历，以此提升手指的触感、对素材的感觉，以及对于物品强硬度及破坏力的认知。发觉并提升人性的本源感知力，是小朋友的特权。小时候习得的这些感知力，在使用数码设备，或是将来创建数码服务及应用时也会很有用吧。

今年秋天将举办首届 Digital Kids Summit（http://digitalkidssummit.com），主题为对儿童数码的思考。未来，针对小朋友的数码设备、应用和服务，很让人期待。最后，我们介绍一下 2013 年 6 月 13 日至 16 日这 4 天，在国际会展中心举办的“东京玩具展”中的儿童数码玩具吧。



预装教育应用的儿童平板。硅胶外套可以完好地保护设备。（MEGA HOUSE）

gadget 1 鸟型音乐屋

*截至撰写本文时，网站未知

手转式的音乐盒

鸟型音乐屋是一个手转式音乐盒，使用纸质曲谱专用打孔器将相应的音阶打孔，然后手工转动把手，用空气的力量使笛子演奏音乐。空气通过笛子的洞可以演奏不同的音符，仿佛是小鸟在歌唱。音乐屋不需要电池，也不是数码类型的玩具，但不知为何，这个音乐屋会让人联想到以前电脑上使用的打孔卡，所以在这里也介绍一下。



gadget 2 AutoMee S

http://www.takaratomy.co.jp/products/automee/

清扫机器人

AutoMee S 是一个超小型清扫机器人，可以清扫智能手机、平板电脑等。AutoMee S 颜色共有 4 种，价格为 1575 日元（含税）（约人民币 90 元）。直径 7cm，比年成人的手掌略小，它可以在智能手机上纵横穿梭，将指纹、皮脂污垢等毫无遗漏地擦走。当它到达智能手机或平板电脑设备的边缘后，会自动改变方向不会掉下去。智能手机的清扫时间大约为 4 分钟，iPad 等平板电脑大约为 8 分钟。



gadget 3 声音算盘

http://daiichi.c-syoku.com/shop/kyouikusyouhin/voiceabacus.html

数码声音算盘

声音算盘是学习算盘技巧的教育小工具。它支持日语和英语，可以用来学习数字、数值以及算法，也可以帮助不懂算盘的外国人来理解算盘的结构原理。这是一个将传统与数字化结合，用数字化实现传统新价值的优秀案例。定价：3990 日元（含税）（约人民币 230 元）。声音算盘是由知名老牌算盘厂商出品的。



gadget 4 KEROMIN

http://www.keromin.com/

玩偶型乐器

KEROMIN 外形是一只青蛙，实际上是 TERUMIN 的腹语术型电子乐器。像手套一样戴在手上，将嘴巴部分一张一合，就可以进行演奏了。定价为 49,875 日元（含税）（约人民币 2900 元），作为玩具价位稍高，但如果考虑到它也是一个乐器的话，就能接受了。控制青蛙的嘴巴，还可以变换音阶。与第一代的版本相比已经改进了很多，并且还搭载了 MIDI 输出。演奏不仅可以使用一般乐器的声音，还可以使用录制下来的真实青蛙叫声。



文 / 结城浩（Hiroshi Yuki）
http://www.hyuki.com
Twitter: @hyuki
译 / 成勇

结城浩的再发现随想 3



Consistent

Consistent——一致

一致是什么

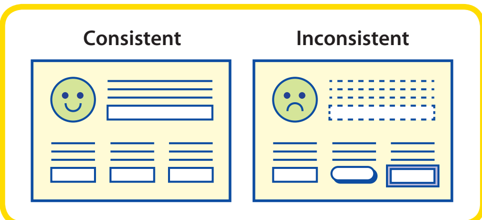
consistent 是形容词，意思是“一致的，统一的”。名词为 consistency，意为“一致性”。反义词 inconsistent，即“非一致的”。

把集中的某些事物进行评价时，一致性是一个指标。

比方说，我们评价网页设计时，会说“设计缺乏一致性”，这也正是我们要讨论的一致性问题。

究竟是在什么情况下，人们会评价网页的设计缺乏一致性呢？常见的情况是用不同设计表现相同的内容。例如，左边的按钮是平面设计，可右边却变成了立体设计，我们会明显感到这两个设计缺乏一致性（图 1）。

▼ 图 1　不一致的网页设计



有时候，我们还会说某个人的代码写法不统一，这也属于一致性的问题。

究竟在什么情况下代码会不一致呢？常见的情况是使用不同代码来表现相同的内容。如果用三个不同的标识符 btn、button 和 b 混合表示按钮，就会让人感觉明显缺乏一致性（图 2）。

按照规则实现一致

判断是否具有一致性时，需要考察多个对象。如果仅有一个对象，就不存在一致性问题。英语短语 be consistent with……就明确说明了一致性是需要具备参照对象的。

前面的例子介绍了按钮设计和标识符写法的不一致。所谓的不一致必然是指两个以上的对象产生了冲突（图 3）。

想要保持一致性，需要遵循一定的规则。这些规则有时候被叫做标准、手册、方针……叫法不同，但都是为了保持一致性而制定的规则。

▼ 图 2　不一致的代码写法

Consistent

```
...
button_reset = ...
...
button_send = ...
...
button_back = ...
...
```

Inconsistent

```
...
btn_reset = ...
...
button_send = ...
...
b_back = ...
...
```

3 结城浩的再发现随想

比方说，规则规定使用平面设计，那么网页制作者就会按照规定设计。或者规则规定按钮的标识符是button，那么程序员也会按照规定写代码。

要想让大规模的项目保持一致性，就需要规则来约束，并依照规则进行检查。

规则本身的一致性

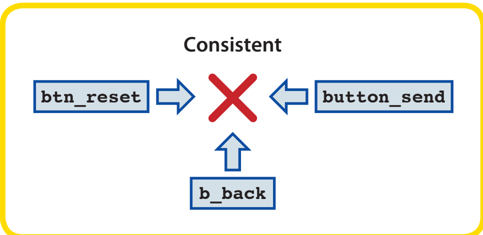
规则也同样如此，若太多就难以保持一致性。

以网页设计为例，很难逐一针对其形态、颜色、动作等内容来制定规则。如果编码规则过细，虽然便于开发人员操作，但却给规则制定者带来了麻烦。

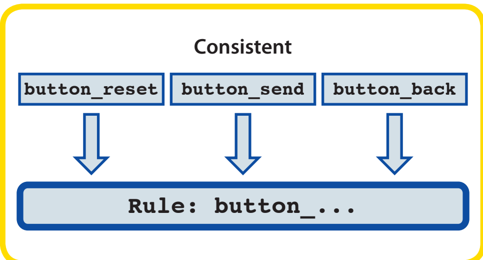
一致性和制作者

不论是网页还是代码，按照一致性制作大规模项目时，我们会感到规则的约束力，即“这不是随便编写的，而是有规则可依的”。进而，还能感受到制作者的存在。实际上可能有多位制作人参与该项目，但是如果大家遵从同一规则分工合作，产生的项目就会具有一致性，让人觉得这是一个人完成的。

▼ 图3　判断一致性需要考察多个对象



▼ 图4　保持一致性需要遵循相应的规则



人类是一致的吗

那么，我们人类具有一致性吗？可以说，有些可以预测到的行为就是一致的。比如，某人一说起电影来就滔滔不绝，却对时尚不感兴趣，那么他的爱好就是保持一致性的规则。

有些事物的一致性隐含在深层次中，可能乍一看看不出来。例如，我有一个朋友对折纸、建筑和昆虫感兴趣（图5）。

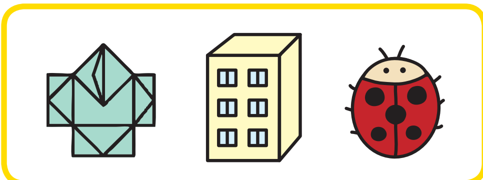
从表面看来，折纸、建筑、昆虫三者之间好像没有什么共同之处。但是，仔细一下你会发现这个朋友的兴趣点在于形状。

这个人喜欢制作形状也喜欢观察形状，在这一点上，折纸、建筑和昆虫有着很好的一致性。

◆ ◆ ◆

你发现自己身边有什么具有“一致性”的东西了吗？它的背后有什么规则吗？请你一定要留心思考哦！

▼ 图5　虽然看起来没有一致性……



结城浩（Hiroshi Yuki）

生于1963年，日本资深技术作家和程序员。在编程语言、设计模式、数学、加密技术等领域，编写了很多深受欢迎的入门书。代表作有《数学女孩》系列、《程序员的数学》等。

作者网站：http://www.hyuki.com/

enchant
～激发创造力的魔法～

文/清水亮（SHIMIZU Ryo　Ubiquitous Entertainment Inc股份有限公司）　译/印勇刚
URL http://www.uei.co.jp

第4回　乘风破浪正此时——一亿人程序员的国家计划

我们在秋叶原创立了ARC（阿克）——一个研究、教育组织。作为一个组织机构，首先需要明确的是，它所面对和需要着手解决的课题。对ARC而言，当下的目标（也就是课题）是如何积极推进新游戏引擎enchant.js，和游戏开发基础环境9leap两者的普及活动[①]。

但是，ARC作为一个崭新组织，还有不少问题需要解决。

开放空间的难处

ARC坐落于秋叶原UDX内部的高级公寓。选址于此主要是因为，东海大地震之后此处房价走低，且面积合适。ARC的办公室位于28层，放眼望去，秋叶原和东京晴空塔（东京Sky Tree）的全貌一览无余，无比壮丽。

可是，随着UEI公司[②]规模的日益扩大，从信息安全角度来看，这种从民用公寓改装而来的ARC办公室，及利用VPN连接的方式，越来越难以为继。

ARC的人员构成上，我的决定是，除数名骨干UEI正式员工之外，其余基本成员全部由学生组成。而且，这些学生可以自由地带着他们的朋友来到ARC办公室现场体验。这样的一个开放空间，也需要一个信息安全层级相对较低的场所。

但是，信息安全层级下降，构成了信息管理与交流方面的障碍，最终导致ARC办公室被公司其他部分的网络孤立，成为了一座孤岛。

刚开始的时候，伏见辽平有时会把他的朋友们带过来，我也常常在这里约见被介绍而来的学生们，和他们聊天，开放性地使用ARC的空间。不过，对于那几位UEI的正式员工来说，ARC就不是一个好的工作环境了。为了要打印一份会议议稿，他们不得不特地奔波十几分钟回到UEI公司总部。而且，会议中可以谈论的内容，也会因为会议场所是在公司总部，还是在ARC办公室而发生很大变化。所以，公司的



在秋叶原高层公寓起步的ARC，以伏见辽平为核心的大学生们在这里彼此交流着意见和创意，期待着新的发明由此诞生

① enchant.js是UEI公司开发的基于HTML5的游戏引擎，免费开源。leap是Leaders Education Achievement Program（领导者教育成就计划）的简称。落实在具体层面，可以访问code.9leap.net和www.9leap.net这两个网站，其主要作用是，为登录者提供一个在线环境，用来学习和使用HTML5 enchant.js创建自己的游戏，以及获得游戏玩家的及时反馈。——译者注

② ARC是UEI公司的一个部门，本文作者清水先生，是UEI公司的负责人。——译者注

正式员工对此非常不满。

“为什么 ARC 是必不可缺的？”“这个部门，究竟在做些什么啊？”“ARC 这个部门，到底是为了什么目的设立的？”

终于有一天，一向娴静文雅的辻秀美，冲我大发脾气，宣泄不满之情。一切如同突如其来的火山喷发，使得周围的人都大吃一惊。此时此刻，我也终于明白，不能再把大家对 ARC 的疑问与困惑，若无其事般地隐藏起来了。

我曾一厢情愿地认为，ARC 存在的必要性和意义，就算 UEI 公司内部的其他人都不理解，辻秀美是可以心领神会的，因为她从还是学生的时候，就开始跟随我周游世界共同工作了。然而，从目前的情况看来，我的这种一厢情愿，实在有些过于乐观了。

目前，有很多正式员工对 ARC 怀有不满。“开什么玩笑，简直像没长大的孩子一样！”，很多公司正式员工一定不屑地认为，放入 ARC 部门的人，都是被特别优待的娇宠儿。“把钱轻轻松松地散给学生们，还发放着毕业后也无需返还的奖学金，这种做法究竟能给公司带来什么利益？”这种想法弥漫在公司正式员工的心头。为此，我不得不把 ARC 的正式员工们聚集起来，耐心地进行说明。

事业和教育

我为什么会在公司内部设立一个专门负责教育和研究的部门呢？

诚然，无论从公司内部还是业界常规的角度来看，ARC 都是非常特殊的一个部门。在 ARC 的主要事业之一，即 9leap 方面，只要学生可以获得高级认可，那么就可以无偿得到最新的 PC 机，而且参加 9leap 也是完全免费的。ARC 的另一个主要事业 enchant.js 同样也是完全免费的。

UEI 公司的正式员工，必须是研究生毕业（或者拥有与此相当的才能）才拥有有加入 ARC 的资格，而 ARC 的其他成员，则全部是在校大学生。当然，这些大学生也都是通过严格的面试



平日里的 ARC。通过动态 IP 地址，学生们想坐哪就坐哪，这也是公司正式员工不满的地方。照片右边的是在 ARC 和公司总部疲于奔波的辻秀美

和笔试进行选拔的。这些学生，从 ARC 领取薪水，却并不负担明确的工作指标，唯一要遵守的规则就是——把自己想做的东西提交出来。ARC 给这些学生提供最新的 PC 机，以及所需的各种设备。

制作免费软件；像散财童子一般无偿地提供 PC 机；花钱雇用学生却不要他们做具体的工作。设置这样的部门，究竟是为了什么啊！？

面对这样的疑问，我的回答是：

设置 ARC 的所有目的，就是“教育”。

作为企业，并非总是可以获得优秀的人才。传统的新生集体招聘，对中小企业来说，很难吸引到优秀的人才。目前，支撑着 ARC 的辻秀美和近藤诚两位，便是明证。他们是通过兼职的形式接触到了 UEI 公司，在几年的时光中，他们充分了解了 UEI 公司的情况，然后才正式加入公司。同样，公司方面，也通过这个过程，充分了解和熟悉了他们的为人、性格、才华等，然后才放心地正式签约。这是只有通过学生时期这个充满感情的阶段的酝酿，才能产生的独特信赖关系。

新人教育对 UEI 公司至关重要，新产品开发人才、以战略眼光提升销售额的决策人才，都需要在新人教育中培育。从创造新产品的角度来看，让集体统一招聘的新生，随意地负责项目策划，这种方式根本不可能发现那些核心且稳定的优秀策划人才。也就是说，对于技术性企业，在项目策划、战略立案时，只停留在技术层面是完全不够的，负责人必须拥有良好

的构思和视角，来洞察世间微妙，读懂人心深处，并且将抽象的欲望以恰如其分的形式鲜活地表达出来，并予以把握。

而这样的构思和视角，仅凭单纯追求技术是无法获得的。哪怕中途放下技术，硬着头皮来学习也无法获得。教养、构思、技术三者兼备，才可被称作优秀的策划者。

此外，没有人生来就具有战略性的长远眼光并能将之转化为现实。世上也很少有这种实践的机会。对我们UEI公司来说，如果无法感知这种战略性的机会并制作出相应的产品，那么公司就要危在旦夕。为了避免这样的悲剧出现，日常就需要加以训练。

中小企业的竞争力，体现在从业界标准来看，它所拥有的人才有多少独特之处。

当大家认为，我们公司里面聚集着独特人才时，大家就会把必须依赖这些“独特才能”的业务交给我们公司。这样的业务，就算是价位较高，也会因为其专业性而不愁没有客户。这就是我们所考虑的价值创造。

如果大家认为公司里面尽是一些寻常人物，那么公司面临的就只可能是价格竞争了。如今，随着印度和越南外包开发业务的蓬勃发展，不以“拥有独特技能的人才”为武器的软件公司，将难以发展和生存下去。

如何才能创造出（培养出）公司自身所独具的人才呢？答案只有教育。而ARC设立的理由，就是专注于教育。

水滴与河流

即使我说明至此，还是没有得到ARC正式员工的理解。

“清水先生，你这样做，难道不是仅为了个人不切实际的理想，在浪费着公司的金钱吗？”过秀美和近藤诚目不斜视地看着我说道。

“公司，如同流动的河川。而员工就像推动河川流动的无数水滴，谁也无法在其中永生。而且，谁也无法阻止水滴通过支流流向其他的河川。以前，当你们以兼职的身份来到UEI公司时，如果我当时就说‘既然来兼职了，将来就留在UEI公司工作吧’，那么你们还会接受那份兼职吗？我想你们一定会心怀疑虑，会觉得‘世界这么大，我还想好好看看呢’，或者想‘这个公司当真是个靠谱的公司吗？’等等。”

“啊，确实可能会有这种疑虑……”

“公司唯一可以做的，就是在人才发挥才能时，为其提供最佳的环境。这种环境包括可以信赖的工作同伴、前辈、指导者，包括可以激发释放其自身潜力的组织机构。这些就是公司可以提供的一切。当看到、体会到这一切之后，有权做选择的，不是我们，而是‘他们’，宛如当时的你们。”

在听了我的解释之后，过秀美和近藤诚表面上并没有表示出完全认可我的样子，但从那之后，他们俩就又开始默默地努力工作了。最终，我也放弃了彻底说服他们的念头，我在心中祈祷，总有那么一天，他们能够理解吧。他们不是那种可以把不满情绪隐藏不露的人，但不管怎样，ARC这艘船已经起航，驶入广阔的大海，剩下的只有前行。

一亿人程序员的国家计划

虽然UEI公司内部还有不少争议，但ARC部门启动发展的新事业——enchant.js和9leap，可谓一帆风顺，蓬勃向上。每个月举行的9 Days Challenge活动，成效卓著，而且ARC的事业，也很快吸引了公司以外其他各界的目光。

ARC的核心事业，是由东京大学的伏见、庆应大学的伊藤、电气通信大学的杉本所推进的9leap项目。9leap配合东京大学的田中谅所开发的enchant.js（使用enchant.js可以简单地开发出游戏），可以快速上传并且及时收集到用户的真实反馈，因而广受好评。这种以学生为中心的组织方式的优势，也很快显现出来。他们勇于开拓、好胜积极，就算面临着不少的困难，也都毫无惧色、奋然前行，挑战着各种新的可能。

不久，《周刊ASCII》《月刊BestGear》上也开始连载相关的内容。许多业余编程爱好者

也开始接触 enchant.js 和 9leap 了。

伴随 enchant.js 的公布，我们也相继在东京、仙台、札幌、大阪举办研习会。自然，这些都是免费无偿的。随后，在冲绳、九州、福井、以及任何有需求的地方，我们都免费派遣讲师过去。而且，举办研习会的时候，作为开发者的伏见、辻秀美以及我本人，也时常一起出席。

有时，我们的脚步也会迈向整个世界，比如，积极参加了世界上首次举办的HTML5游戏大会onGameStart。在大会现场，我们用蹩脚的英语宣传了 enchant.js。在我们准备的宣传手册中这样写道：

在全世界，无论您身处何处，我们都免费向您传授制作游戏的快乐。

由此开始，我们启动了全世界学校范围内的enchant.js宣传巡回活动。

enchant.js是款开源免费软件，不能产生任何的销售利润。这样的宣传巡回活动，究竟有什么意义？我想一定有抱有这种疑问的人。

对于这样疑问，我的回答仍然是：教育。

本来，所谓的教育，应该是教育者传道解惑的同时，学生也自觉自发地努力向前上进。可是，不知道从什么时候起，学生丧失了源于自身的求知欲和探索心，而是在周围的压力下，或者在教育者的压力下，才硬着头皮勉强地继续学习。我总觉得这种情况，早已偏离了教育的本源。

教育应该回归本源。传道解惑的同时，学生应该激发或保持自身的求知欲和探索心。学习的东西，如果正好和本身的喜好相吻合，那么这些体验、学习的体验，就有了支付金钱的价值。这种感觉，如同迪士尼乐园的门票或者电影院的电影票那样。

而当学习的喜悦还没有被充分发现和感知的时候，我们首先应该让学生们逐渐感知到这种学习的喜悦。这就是我们无偿地奔波全国各地，举行研习会的原因。

那么，让越来越多的人，通过这种无偿的方式了解和熟悉了enchant.js 之后，会有什么样



为普及 enchant.js，在全国各地举办的研习会。照片为在crypton公司大力协助下举办的札幌研习会



ARC 参加博客媒体 wise9 的采访旅行



波兰HTML5游戏大会onGameStart的会议现场。照片里面可以看见 enchant.js 的 Logo 标志

结果呢？我们会得到一些什么呢？

结果就是，我们将得到亲切感。

如同甲子园中的棒球少年与教练、队友构筑起的亲切感一样，我们也希望和程序员们之间拥有这种亲切感。

亲切感同样也是金钱无法买来的珍贵感情。编写软件作为一种教育事业，如果成为一种扎根在喜悦基础上的兴趣教育，而不是扎根在烦闷和不安基础上的应试教育，那么将有多少人会对编写软件充满了亲切感啊。

为此，被称为“一亿人程序员的国家计划”的活动开始了。让编写程序变得更加简单，让编写程序变成一种容易产出亲切感的美好事物。这个目标前进的第一步，就是 enchant.js 和 9leap 。

我所偏爱的键盘图鉴

文 / 造梦有限公司　滨野圣人
khiker.mail@gmail.com
Twitter：@khiker
译 / 卫昊

第 4 回

没有按键的键盘？

orbiTouch & COOL LEAF



图片 1　orbiTouch

引子

上一期，我们介绍了人体工学键盘的最强者 KINESIS Contoured Keyboard 以及 Maltron Keyboard，本期将继续介绍人体工学键盘。人体工学键盘有许多种类，拥有奇特外形的也不在少数。这次将介绍其中最奇特的无按键键盘 orbiTouch。同时，也将介绍同样没有按键的 COOL LEAF 键盘。

orbiTouch

orbiTouch（图片 1）是 Blue Orb 公司生产的人体工学键盘，是一款著名的无按键键盘。

特点

oribiTouch 的特点除了没有按键，键盘自身也相当大。虽然拥有特殊的外形，但操作系统还是会将它识别为普通的英文键盘。

键盘上有两个突起的操作碗，可以进行上下左右八个方向的移动（以滑动的形式）。仅通过抓住两侧的碗就可以进行操作，手指和手腕几乎不需要参与。即便是身体上有严重障碍的用户，也可以随心使用这款键盘。

操作上，与格斗游戏的出招命令十分相似，但还是有些许区别。左侧的碗向八个方向中任意一个方向滑动后，再滑动右侧的碗就相当于输入了某个按键。

为了辅助操作，左侧碗的各个方向用红、黄、绿进行区分（图片 2）。在右侧碗各个方向标有当左侧碗处于相应颜色时，滑动右侧碗会输入的字母（图片 3）。例如，当左侧的碗滑动到左边时（红色的部分），右边的碗向左滑动后，就会输入字母 a 了。

Shift、Ctrl、Alt 之类的修饰键也可以进行操作。orbiTouch 拥有模式的概念，左侧的碗向上滑动两次进入 Shift 模式，在此状态输入普通文字时就相当于按住了 Shift 进行输入，Ctrl、Alt 也是一样的操作方式。

模式功能中也包括鼠标模式。左侧的碗向下滑动两次就会进入鼠标模式。进入鼠标模式后，左侧的碗可以进行鼠标的左键单击以及右键单击，右侧的碗则可以用来移动光标，不过，鼠标滚动的功能并没有配备在该键盘上。

关于如何移动会输入哪个字母，在 orbiTouch 的 CheatSheet 上进行了标准化规范（图片 4）。

由于像这样学习成本很高，操作碗向八个方向移动也很费

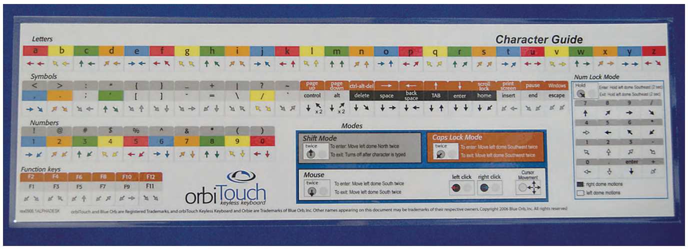


图片 2　orbiTouch（左侧的操作碗）



图片 3　orbiTouch（右侧的操作碗）

orbiTouch & COOL LEAF



图片4　操作命令CheatSheet

力气，所以想拥有和普通键盘相当的操作速度是很困难的。不过，为了进一步辅助学习，除了之前提到的CheatSheet之外，此款键盘还附赠了说明书以及包含练习软件的CD。

键盘接口虽然是PS/2，但是笔者购买时附赠了PS/2到USB的转换接口。最近，此款键盘还发布了使用蓝牙技术的无线版本。

购买方法

orbiTouch在其官方网站[1]及美国amazon.com均有售。只不过，截至笔者购买时，售卖网站仍然无法发货至日本。至于日本代理店，据笔者所知是没有的。另外，拍卖网站也只是偶尔有拍卖品在进行拍卖。

最简单的购买方式是通过海外代购者以及amazon.com代购者进行海外代购。笔者就是采用这种方式购买的。代购费用也不尽相同，建议大家好好调查一番后再购买，此款键盘的售价大约在399美元左右（折合人民币约2429元）。

COOL LEAF

COOL LEAF是来自Minebea公司的全平板键盘。

特点

没有按键，整体采用镜面平板，是这款键盘的特点。虽然外表看起来十分独特，但操作系统还是会将它识别为普通的ＵＳＢ日文键盘。

键盘连接至电脑前，镜面上不会显示任何内容（图片5）。接入电脑后，按键将以LED背光的形式浮现出来（图片6）。触摸按键即可进行键盘输入。触摸时会发出“哔”的提示音。COOL LEAF的触摸识别是静电电容式，而非压感式，所以带着手套触摸是无法被识别的。

另外，由于键盘采用整体镜面面板，清扫起来非常简单，仅用布擦拭即可。不过反过来看，不得不说键盘的输入有一些困难。基准键位比较难找，没有很好地练习过盲打的话，使用起来将会相当困难。

根据笔者的感受，该键盘代替普通的键盘来使用还是很困难的。笔者认为这是一款在对清洁度有要求或有其他特定用途时使用的键盘。

购买方法

以前，在Amazon等日本在线商店才可以购买，现在可以购买到的地方好像变多了。在秋叶原的各个店面寻找一下说不定就会买到。价格方面，笔者购买时大约为2万日元（折合人民币约1229元）。

◆　◆　◆

本期介绍了没有按键的键盘，像这样风格怪异的键盘还有好几款，感兴趣的读者不妨去寻找一番，说不定也是一件十分有趣的事情。



图片5　COOL LEAF（电源关闭）



图片6　COOL LEAF（电源开启）

① http://orbitouch.com/

创客在行动

Maker Faire中的3D打印机

文/坪井义浩　TSUBOI Yoshihiro　ytsuboi@gmail.com　Twitter @ytsuboi　译/苏祎
协助：switch science股份有限公司　http://www.switch-science.com/

上期Maker Faire Bay Area的介绍中，有许多与3D打印机相关的展出。本期Software Design的特辑1也是“3D打印机拓展未来的理由”，所以我们此次也将介绍上期未能详述的3D打印机相关内容。

会场所见之中，大型的Ultimaker Robot给笔者留下了深刻印象（照片1）。不知为何，它被安置在会场角落的食品柜台上，但它的尺寸确实非常巨大，高度竟有20cm。

本期特辑1中负责撰写第3、4章的山田齐先生，选择了Ultimaker 3D打印机的其中一个原因就是，去年我们在Open Hardware Summit中拿到了Ultimaker Robot打印品，山田齐先生非常满意其品质。而且打印出机器人大型身姿的输出过程，也让人难以忘怀。

123D Catch 照相展区

大型3D CAD厂商Autodesk公司设置了上身3D模型拍摄展区，以展示并推广其123D Catch[①]技术（照片2）。这个展区非常受欢迎，很多人在排队等待拍摄，笔者也参与了拍摄。

展区内有大量的单反相机排列着，从各个角度对人的上半身进行一次性拍摄（照片3）。

拍摄好的数据会在日后以电子邮件形式发送，文件类型为Autodesk 3D model扩展名.3dp的文件，此类文件可以通过电脑或iOS的123D Catch应用打开。

虽然有人收到了拍摄成功的3D照片，但笔者的照片数据如照片4所示，真是让人遗憾。这是我通过iPhone版的123D Catch打开文件，失望之余时的截屏。当然，这是3D图像数据，通过手指操作就可以旋转照片。

既然是3D数据，笔者就希望能通过3D打印机将自己的半身像打印出来看看，但非常遗憾没能实现。

Autodesk公司从之前开始就和Makerbot公

① http://www.123dapp.com/catch

▼照片1　大型Ultimaker Robot



▼照片2　123D Catch 照相展区



▼照片3　展区内部



▼照片4　笔者的3D照片



司进行业务合作，在该公司的展区放置着多个Replicator2。以往Autodesk公司的软件都是非常昂贵的专业产品，但近年也推出了针对模型爱好者的免费软件。除去以123D Catch为开端的123D系列以外，还公开了Auto CAD WS以及Inventor Fusion等功能缩减的免费版本。若是OS X用户的话，在App Store中搜索Autodesk就可以找到这一系列的软件。

另外，在Maker Faire商品售卖处，有Replicator 2和Ultimaker等3D打印机现场售卖（照片5）。Replicator 2平时须在网上预约才能买到，所以现场的销售让笔者很动心，差点就买下了。但想想要带回国会非常麻烦，所以还是忍住了。很多携家带口的美国家庭热情购买使得3D打印机卖得非常好，众多人扛着它回去的样子让笔者非常惊讶。

SandBox

SandBox[②]是以尚在研发中的粉末为材料的开源3D打印机（照片6）。虽然网站上还没有很详细的资料，但是可以大致了解到SandBox的打印方法是，使用强力胶将粉末加固并输出。大概是使用了本刊特辑1第2章中也介绍到的粉末固定积层法吧。材料的粉末也可以选择石膏、陶瓷、木材、玻璃、砂糖等各种材料。

SandBox预定于今年秋天发布，并且代码也准备在GitHub公开。若有兴趣的话可以在网站上登记一下邮箱地址。

虽不能确定是否为SandBox输出打印而成，但SandBox网站上介绍的UCB Department Architecture的3D打印成果也被展出了，该作品非常很吸引眼球（照片7）。UCB这个词，可能只有熟悉Unix的读者会对其有所反应，其实它是指孕育了BSD（Berkeley Software Distribution）的加州大学伯克利分校。

Type A Machines

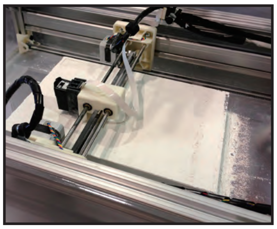
Type A Machines Series 1[③] 3D打印机，在美国已经被大众熟知，但其影响力还未延伸至日本（照片8）。展示的打印机上有写着BETA，但这个打印已经在网上进行销售，售价1695美元（约人民币1万元）。

Type A Machines还开发了Series 1的改良机型，此款新机型Series 1 Pro也在展出中（照片9）。新机型将框架更换为铝质，速度和可靠性都得到了提升。Series 1 Pro目前仍是开发中的工程概念机，此机型将对应Wi-Fi连接，并计划在今年下半年正式发布。

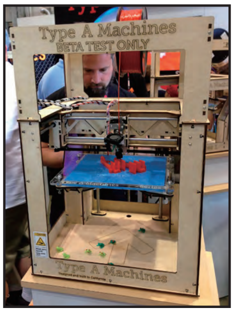
② http://sandbox3dp.com

③ http://www.typemachines.com


▼照片5　商品售卖处的现货


▼照片6　SandBox


▼照片7　UCB木制打印的展示



▼照片8　Type A Machines Series 1

3D monstr

这也是一款笔者未曾见过的3D打印机，所以对其充满了好奇。3D Monstr[④]是以高精度部件和改良型Exstrooda（详情请见本刊特辑1第2章）为卖点的3D打印机（照片10）。至今还没有公布详细信息，所以无法了解到细节。

3D Monstr似乎计划通过名为Kickstarter的募资网站筹集资金，所以展出打印机的上方也有与此相关的展板。但Kickstarter上的具体活动貌似还没有展开，所以这次可能是为募资打响知名度进行的前置展出。

Afinia H-Series

虽然在美国的*Make*杂志及其博客中经常会看到此款3D打印机，但Arinia H-Series[⑤]的价格为1599美元（约人民币9700元），属于低端分类。这使得笔者对其打印完成度略微担忧，而且它与其他打印机相比打印速度也慢，所以笔者之前对它也没有太多关注（照片11）。

④ http://3dmonstr.com
⑤ http://www.afinia.com

笔者访问了其展台，想看一下此款打印机的品质能到何种程度。结果，在查看了其样品以后，惊奇地发现其实际品质远好于笔者想象中的效果。（照片12）。

笔者在观看打印样品时，旁边一个人的话给笔者留下了深刻印象。他说自己拥有Afinia，虽然打印速度比较慢，但使用方便，打印品质也比较让人满意。在看到此次的打印输出样品之前，笔者从未想要Arinia H-Series，但见识过以后，觉得这款打印机还是不错的。

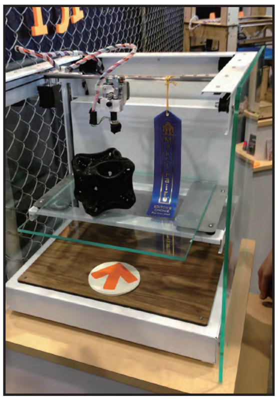
RepRap

RepRap[⑥]是开源3D打印机，但展出的机器是由当地Bay Area的用户群体Bay Area RepRap自行组装的。

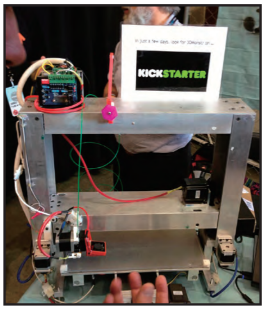
RepRap有多种机型，这里展出的是Mendel 90[⑦]（照片13）以及Prusa i3[⑧]（照片14）。RepRap的相关内容会在本刊特辑1第5章中介绍。

⑥ http://reprap.org
⑦ http://reprap.org/wiki/Mendel90
⑧ http://reprap.org/wiki/Prusa_i3

▼照片9　Series 1 Pro



▼照片10　3D Monstr



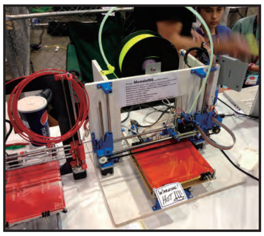
▼照片11　Afinia H-Series



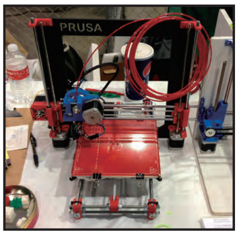
▼照片12　Afinia的打印样品



▼照片13　Mendel90



▼照片14　Prusa i3

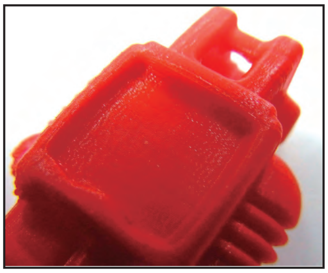


特辑1扩展包　3D打印机关联介绍

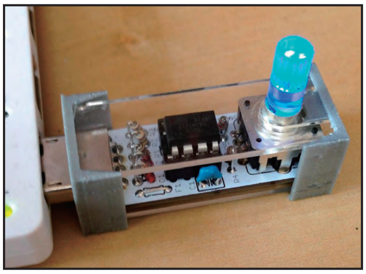
这里扩充了与本刊P17开始的特辑1“3D打印机拓展未来的理由”文章相关联的资料，收集汇总了文章中希望通过彩色照片介绍的内容。

●特辑1　第1章

▼照片15　笔者拥有的Ultimaker Robot的打印成品

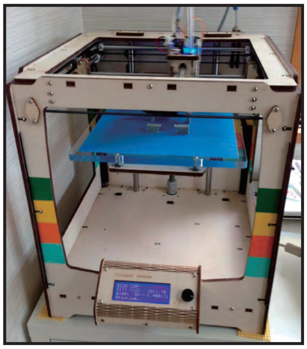


▼照片16　在基板上固定亚克力板外壳的零件

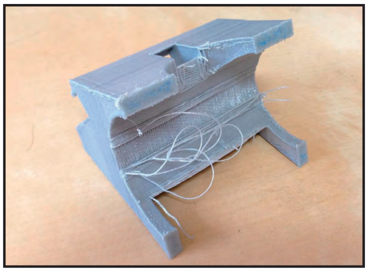


●特辑1　第4章

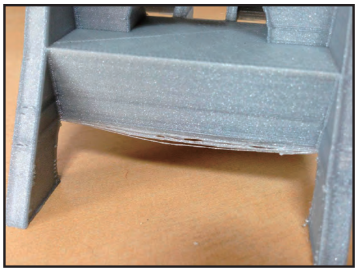
▼照片17　山田先生的Ultimaker



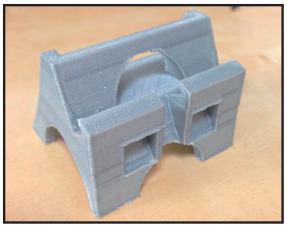
▼照片18　残留的纤维屑



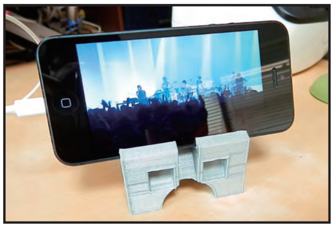
▼照片19　Over Hung造成的树脂下垂



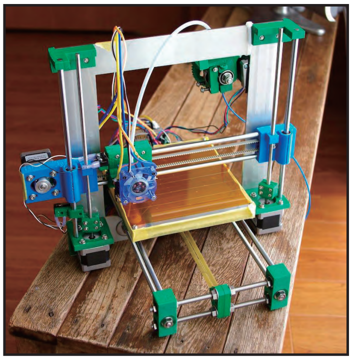
▼照片20　完成的iPhone支架（最终型）



▼照片21　将iPhone5横向放置的样子（最终型）

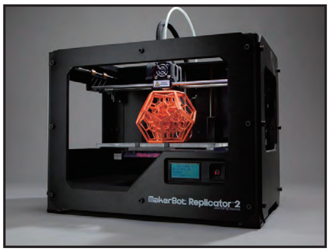


▼照片24　RepRap atom

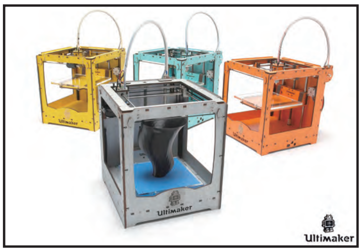


●特辑1　第5章

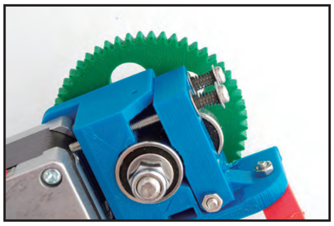
▼照片22　Replicator2



▼照片23　Ultimaker



▼照片25　atom的Exstrooda

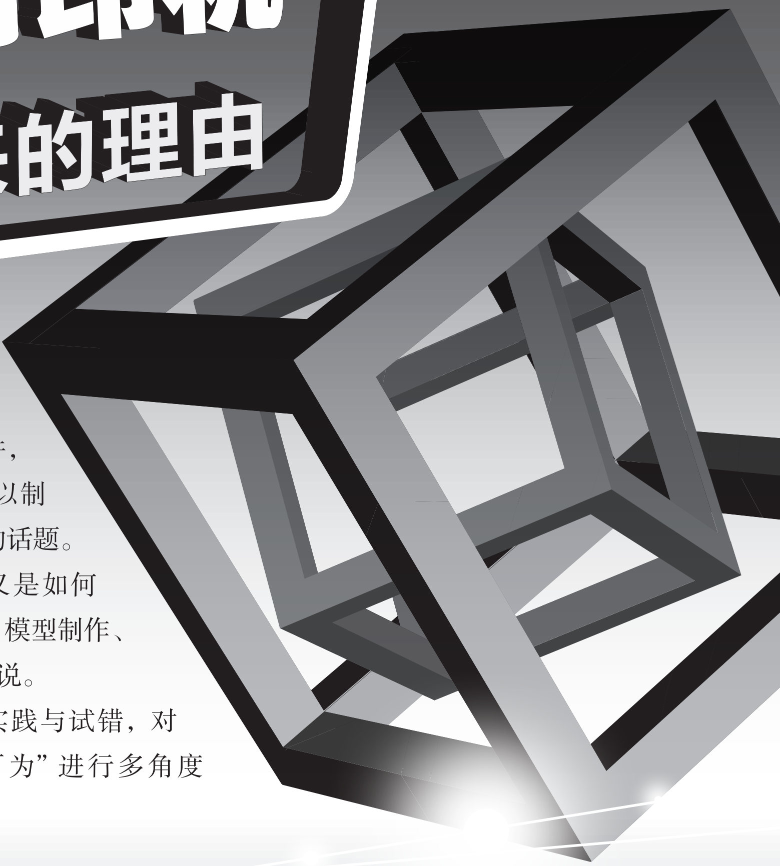


特辑1

从理论到实践

3D打印机

拓展未来的理由



最近，“3D打印机”荣登各种热门话题。可以用来做版面设计编排，可以制作无法购得的零部件，可以制作构造复杂的作品，甚至可以制作枪支……仿佛它已经独占了所有的话题。

3D打印机究竟是什么构造，又是如何打印作品的？本特辑将对具体理论、模型制作、3D打印技术进行循序渐进的详细解说。

我们将立足于实际制作中的实践与试错，对3D打印领域中的“可为”与“不可为”进行多角度分析。

第1章 3D打印机的现状分析

文/坪井义浩 ytsuboi@gmail.com Twitter @ytsuboi
译/kaku

本章将详细解说时下热点3D打印机及其相关内容。除3D打印机外，还将涉及与之竞争的激光加工机和CNC数控铣床、3D打印机的数据格式以及3D扫描等内容。

前　言

近年来，3D打印机以迅雷不及掩耳之势聚集了大众的目光。3D打印机热潮空前兴起，首先是因为其价格进入大众可接受的范围，再者是科技创新使得3D打印机在使用上日趋简便。例如，自Makerbot公司2012年发售Replicator以来，成品3D打印机就开始出现在市面上。之前3D打印机都是以零部件的形式来销售，收到打印机后，要经过很长时间的组装和调试才能使用。

变得廉价的3D打印机

3D打印机，之前多是汽车和飞机制造公司为制作部件而花费数百万才能买到的东西。如今，普通人花费万余元就可以买到了。

拥有很多“3D打印机之匙”专利的3D Systems,Inc.在1984年取得了光造型技术专利，并于1987年首次将光造型3D打印机商业化。3D打印机行业，就如同IT行业，充斥着专利侵权案件，对于新手，是一个非常难以进入的行业。

然而，20世纪80年代取得的专利已经超过了20年的有效期限。2005年RepRap等项目的启动，开启了个人3D打印机的时代。RepRap项目孕育了各式各样的开源3D打印机，同时，这个项目也是Makerbot公司的起点。另外，Makerbot公司起初开源出售3D打印机的元件，也出售了很多打印商品，但近来出售的大部分都是完成品了。时至今日，如果你想购买一台3D打印机，市面上种类繁多的产品，会让你不知该如何选择。

MIT（麻省理工学院）媒体实验室Bits and Atoms所长Neil Gershenfeld所著的*The Coming Revolution on Your Desktop——from Personal Computers to Personal Fabrication*[①]中也提过，“（几乎）所有的东西都能自己制造的时代”会慢慢到来。这本书中讲述了，从二进制到原子时代的变迁，以及消费者变身为设计者和制造者的时代转变。

另外，著名杂志《连线》的原总编Chris

① 此书已绝版，很难买到，推荐Neil Gershenfeld新装再版的*Fab: The Coming Revolution on Your Desktop--from Personal Computers to Personal Fabrication*（照片1）。

Anderson在畅销书《自由》之后所著的《创客：新工业革命》(照片2）中也这样论述了产业结构将会发生巨大的变化："21世纪的制造业，只要有构想和电脑，任何人都可以从事。"

3D打印机私人化真的会改变当今世界的产业结构吗？用3D打印机真的能制作东西吗？本文将会为读者带来3D打印机的现状分析，若能对各位在3D打印机的选购上提供一点帮助，笔者不胜欣喜。

揭秘3D打印机的"神通"

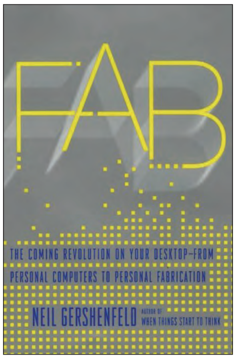
3D打印的"造物神力"

3D打印机制造立体物品，先用树脂或石膏等材料涂抹平面，再从纵向（Z轴）逐层堆叠。

这种逐层堆叠材料的高度叫做积层精度，积层精度的精确程度对成品精细度有非常大的影响（图1）。

近来，消费级3D打印机也有0.3毫米或者0.1毫米规格的积层片。0.1毫米的积层片，即在1毫米中进行10层的逐层堆叠，让人感觉仿佛可以制作出难以置信的高精密度物品。可是，人的敏锐感觉却更胜一筹，仍然可以迅速分辨出制造物是否使用了0.1毫米逐层堆叠。

▼照片1



▼照片2



3D打印的强度

逐层堆叠存在一个局限性，即层与层之间的粘合会变弱。在使用同样树脂的前提下，与树脂溶化后塑型制作相比，3D打印而成的物品，在机械性强度方面存在明显不足。

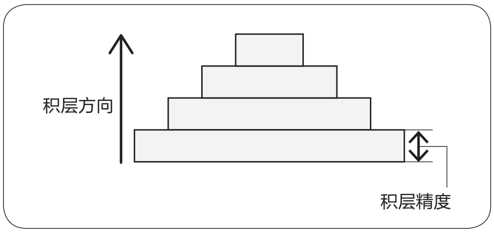
最近有关3D打印机的新闻里，有这样的说法：3D打印机仿佛拥有魔法。略读了新闻里所介绍的相关书籍，发现大家似乎对3D打印机的期望过高了。

笔者认为在这个问题上存在一个很大的误解，即大家认为"只要有3D打印机，就能制作东西"。确实，3D打印机被广泛用于大企业的产品开发流程中，但是，用3D打印机直接量产最终消费品的公司可是一个都没有。

模具也能用3D打印机制作

笔者撰写本文时，正好看到一则新闻，内容是松下使用3D打印机制作量产树脂产品时使用的金属模具。一直以来，这种模具都是由机器切割金属加工而成的，也就是稍后会介绍的数控铣床技术。但松下采用了另一种思路，即

▼图1　积层精度



直接融化金属粉末，并使用3D打印机进行逐层堆叠后，加固成金属模具。

3D打印的时间成本

限制3D打印机量产制品的因素，除了上述的积层精度的和强度外，还有打印所花费的时间成本。像P17照片16中的零件，为了打印出固定2块亚克力板所用的小部件，使用相对高速的消费级3D打印机Ultimaker，大约需要15分钟。完全固定亚克力板，需要一对这样的小零件，这样算来，3D打印机需要不停运转30分钟。如果要制作100对，除去设置打印机的时间，在产出品全部合格的假设前提下，也需要50个小时的连续作战。

3D打印的预算成本

另外，这个部件能卖多少钱呢？一对能卖几百日元（100日元约合人民币6元）算不错了吧。也就是说，即使3D打印机运转一周（一天8小时，一周5天一共40个小时）以上，产出值也只有几万日元（1万日元约合人民币600元）而已。

若使用传统的树脂成型方法，100万日元（约人民币6万元）就能委托中国工厂制造数万个这样的商品。也就是说，平均一个商品的花费在几十日元（10日元约人民币6角钱）左右。这样大家应该理解了，3D打印机并不适合大批量生产制品。

小批量制造最合适

DIY的乐趣不在于大量生产，而是闲暇时制作自己想要的东西。自己制造那些无需大量委托生产的小部件，也正是DIY的乐趣。例如刚才列举的树脂小制品，如果自己很喜欢，但是它已经停产了，这时候，只要花费几十、几万日元（约几百、几千元）和劳力，就能够自己制造出来。

目前来说，虽然3D打印机的时间成本较高，但根据需要，并在合适的领域内选择3D打印，往往会诞生出革命性的创新制造方法。

使用3D打印机制作模型

刚才列举了3D打印机的单体制造实例，若将3D打印机打印出的东西作为原型，使用树脂建模，也可以进行小规模的量产。树脂建模是牙医的常用方法，将树脂灌入通过硅胶制作的原型模具（与原型的凹凸相反，也被称为母型）中，然后加固成型，也称作树脂成型（图2）。

模型制作虽然会费些工夫，但小规模工业生产都会使用这种方法，也是兴趣爱好者制作车库配套元件的首选。树脂建模所需要的材料在模型店和DIY店都有售卖。若想要制造数十个或数百个一样的物品，就可以选择这种方法。

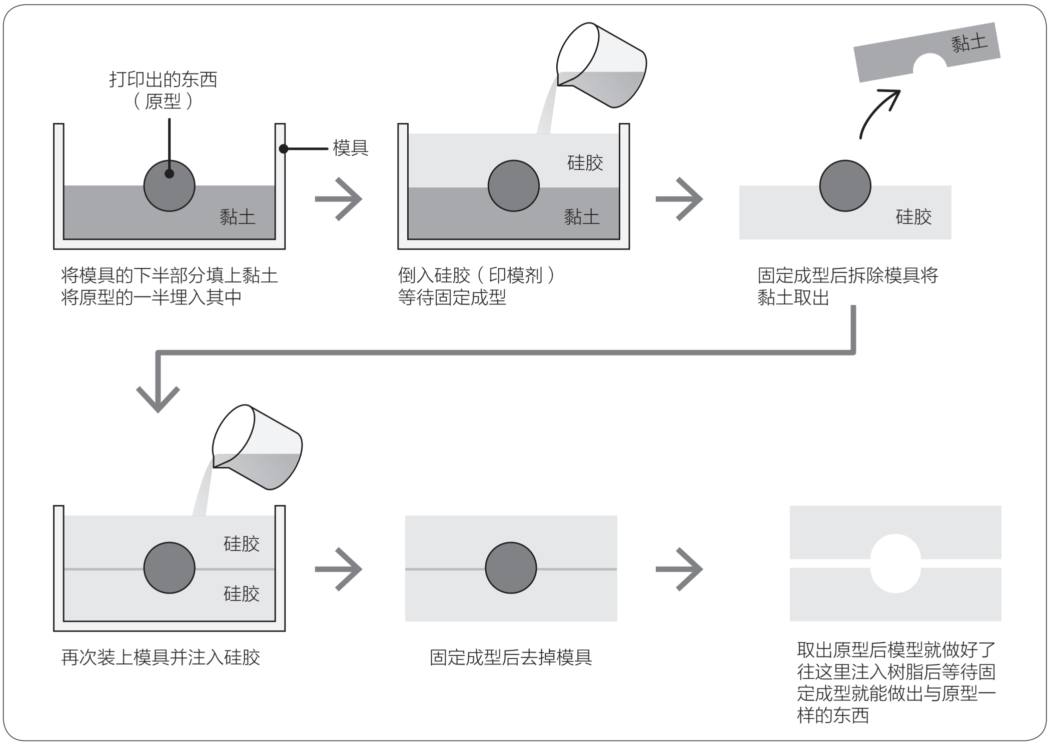
如果选择熔模铸造法，消费级3D打印机也可以制造出金属物品。最近，有篇博客文章[2]让我印象深刻，其内容大致如下：树脂建模是使用树脂制作成型，熔模铸造是使用铸造砂或石膏来制作成型。石膏凝固后，将其高温煅烧，其中PLA（聚乙烯乳酸，第3章会详细解说）材质的原型会进入熔化状态，从而塑造出母型（图3）。熔模铸造之名大概就源于PLA熔化吧。将熔化的金属灌入成型的模型并加固，切开模型，就能得到所需的金属制造物了。

熔模铸造是一种传统的铸造技术，缺陷在于，每制作一个物品就必须消耗一个模型。

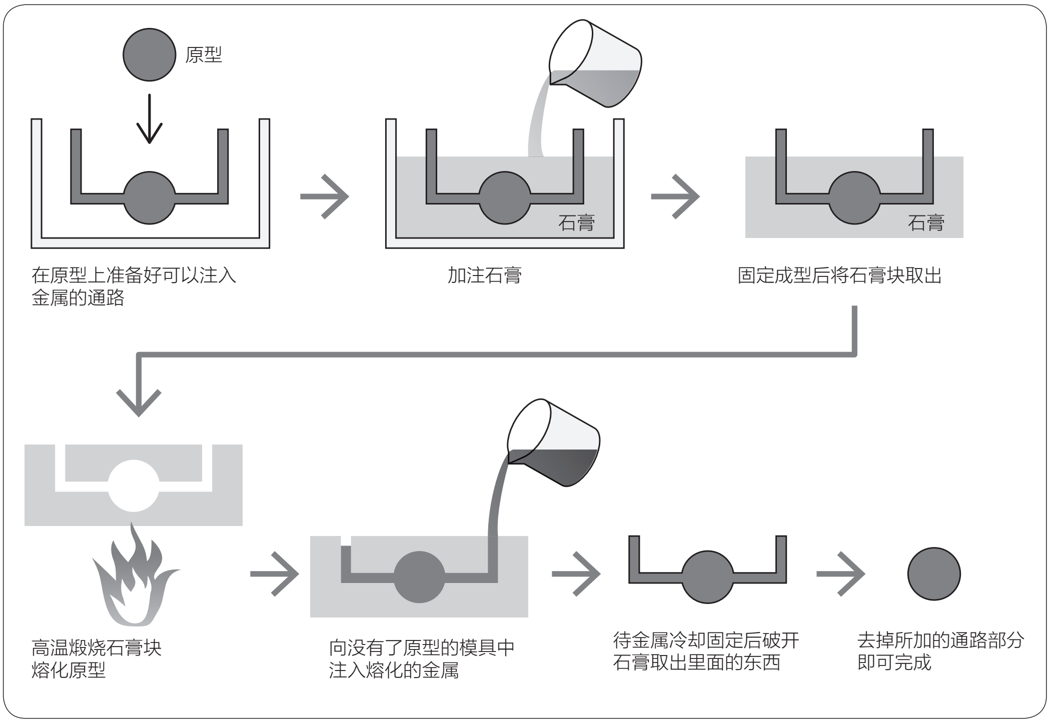
---

② http://3dtopo.com/lostPLA/

▼图2　树脂建模



▼图3　熔模铸造



数字化制造——3D打印机是唯一出路？

将电脑数据转换成实物，只有使用3D打印机一条出路吗？实际上并非如此。我们相对熟悉的方式，还有激光加工和CNC（Computer Numerical Control，电脑数字控制）数控铣床。

激光加工机

激光加工机，是以高强度激光对原材料进行燃烧、熔化、切割等加工的装置。现在，虽说激光加工机已经普及到印章店，但其与3D打印机相异的Z轴方向（高度），调整起来仍是非常困难。另外，关于激光加工机，还存在一个误区，一般家用消费级激光加工机是不能切割金属的，不过，可以轻而易举地切割木头或者树脂板。用镜头聚合激光光纤时，会产生焦点距离等，所以在切割特别厚的材料时，一定要注意斜置横切面。

激光加工机一般用来切割板状物品，然后通过组装平面制造立体物品。最近激光加工机的价位已经降至在30万到50万日元（约人民币1.8万~3万元），所以个人拥有激光加工机也不再遥不可及。

在家中使用激光加工机的困扰是，燃烧材料会产生烟尘，特别是树脂板，加工时会产生相当臭的味道，因此必须安装除烟除臭装置。在购买激光加工机时，请一定不要忘记同时购买除烟除臭设备。

CNC数控铣床

CNC数控铣床，以电脑控制的回转刀对材料进行切削加工。所谓回转刀也就是牙医在削整牙齿时使用的工具。

大型CNC数控铣床，可以加工各种金属制品，不过，一般消费级数控铣床只能加工铝制品之类。如果要进行精细加工，应选用细回转刀，细致削整会导致加工时间变长。精细加工中，由于细回转刀力度较小，所以削整材料时可能会发生弯折。如果进行粗加工，就应选用粗回转刀，加工时间也会相应缩减，当然，加工的细致程度也会随之降低。因此，一般用粗刀来制作大概的形状，然后用细刀细致加工。噪音较大是CNC数控铣床的缺点，若在集体公寓里使用，要注意这一点。

Roland DG公司的iModela（照片4）是著名的消费级CNC数控铣床，价格约8万日元（约人民币4800元），算是比较便宜了。我身边有好几个朋友都购买了这款CNC数控铣床。

3D打印机的数据

之前一直在讲加工机的话题，还没有涉及到3D打印机。要进行3D打印，就需要先准备好3D的CAD数据。不过，这个3D CAD一定要熟练运用才行。最近新出了很多操作简便并且且可以免费试用的相关软件。本特辑第3章，就会介绍一款名为TinkerCAD的3D CAD。

激光加工机

3D打印机使用的3D数据，一般是STL格

▼照片3　激光加工机



式。STL格式由3D Systems公司开发，以三角形的集合体来表现物体，无法表现弯曲等形状。多数3D打印机都是读取STL文件，并将其转换成控制3D打印机运行的信息（G代码）。这种将STL文件转换成G代码的软件叫做切片软件。

大多数3D打印机的控制软件，都能从种类繁多的切片软件中找到可适配的软件。3D打印机对不同算法的切片软件会表现出不同的“喜好”，相同的STL文件，被不同算法的切片软件转换后，3D打印机输出的结果也会产生差异。

切片软件将STL文件转换为G代码，并逐次传送至3D打印机的微型控制器中，从而指示3D打印机输出。所以，大部分的3D打印机在打印过程中，需要保持电脑一直运行。若打印中电脑休眠，打印会被终止。若电脑高负荷工作，也会漏掉处理致使输出品质下降。为了避免这些问题，部分3D打印机将G代码保存在SD卡等闪存中，打印机可以独立读取SD卡数据，并进行打印，从而省去了上述烦恼（照片5）。

3D模型的公开网站

想用3D打印机制造物品，可学习3D CAD建模又路漫漫其修远兮，该怎么办才好呢？实际上，有一个可以公布个人制作的3D打印模型的网站——Thingiverse[3]。

Thingiverse是由Makerbot公司运营的社区网站，上传有数量众多的3D数据，这些3D数据的形式分为STL、3D CAD数据和G代码几种。模型数据的许可上标记了知识共享的标识，使用条件也简单易懂，感觉还是很不错的。Thingiverse上上传的数据包括了3D打印机的配件、日常用品、iPhone手机壳等小配件，展示的物品主题各异又生动有趣，只是浏览一下就会感到愉悦。

③ http://www.thingiverse.com

3D扫描仪

制作3D数据的方法不仅限于使用3D CAD。使用3D打印机，不少人会被卡在3D CAD这一关，但现在，还可以通过3D扫描仪这一廉价手段直接扫描。3D扫描仪原本价格昂贵，但在Makerbot公司开发了Digitizer产品后，开发3D CAD的大公司Autodesk也在iPhone、iPad和电脑平台上发布了免费应用123D Catch，此应用可以通过多图拍摄构建3D模型，这些都推动了3D扫描仪的廉价化。

3D打印服务

即使熟练掌握3D CAD，但对自己手边3D打印机输出的品质仍然不满时，可以使用shapeways[4]服务。将数据传给shapeways就能

④ http://www.shapeways.com

▼照片4 iModela



▼照片5 Ultimaker的控制器



得到3D打印物品。他们所拥有的3D打印机支持全彩打印、金属制作等消费级3D打印机所不能及的功能，可以说相当先进。另外，针对用户上传的数据，shapeways还提供交易平台功能（3D打印和邮寄由shapeways负责）。目前国内的3D打印服务也在逐渐增多。

GitHub痴迷3D打印

本刊读者应该都很熟知GitHub，他们的人现在也在迷上了3D打印。他们的办公室中有一台Makerbot的Replicator 2，这台3D打印机被他们称为Hubot，并与公司内系统相连接。只需要通过Thingiverse等URL下载3D数据，用切片软件切片，就可以使用Replicator 2进行打印了，这个脚本也公开在GitHub之中。

汹涌而来的3D打印潮流对GitHub的影响还不止于此。2013年4月开始，将STL文件上传到GitHub，就可以直接在浏览器中看到3D效果。

希望有一天，笔者也能使用这种方法高效地运行不在身边的3D打印机。

延伸阅读

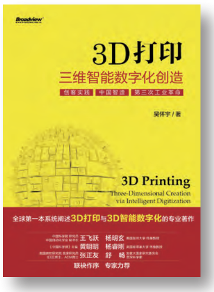


- 首次全面讲述3D打印
- 颠覆传统物理世界

[美] Hod Lipson [美] Melba Kurman 著
赛迪研究院专家组 译
中信出版社

3D打印：从想象到现实

本书讲述了3D打印技术的突破性发展，以及3D打印技术将如何应用于学校、厨房、医院等场所。这本书预测，不远的未来，我们完全可以用电脑把自己想要的东西设计出来，然后进行三维打印，就像我们现在可以在线编辑文档一样。通过电子设计文件或设计蓝图，3D打印技术将会把数字信息转化为实体物品。当然，这还不是3D打印的全部，3D打印最具魔力的地方是，它将给材料科学、生物科学带来翻天覆地的变化，最终的结果是科学技术和创新呈现爆发式的变革。



- 全面系统地介绍了3D打印与3D智能数字化技术
- 详细论述了3D打印和人工智能的现状和未来

吴怀宇 著
电子工业出版社

3D打印：三维智能数字化创造

这是一本以作者原创观点为指导，融汇众多新思想，详细讲解3D打印和3D智能数字化技术原理方法，手把手实战型教学的综合类技术书籍。书中对每一个操作步骤都进行了图文并茂的详细描述，包括实际运作一家3D照相馆的所有技术细节。

本书无论对于国内外的广大普通用户及技术爱好者，还是高等院校大学生及研究生、学术界、工业界、政府产业经济决策层，都具有重要的参考价值。



- 技术推动创新，3D打印未来
- 全彩印刷，带你玩转革命性的3D打印技术

王春玉 傅浩 于泓阳 著
人民邮电出版社

玩转3D打印

本书带领读者了解3D打印技术的起源、发展和现状，结合3D模型构建、3D打印机的操作和使用，介绍了如何掌握和应用3D打印技术、众多的3D打印机的应用案例，并且展望了3D打印给各行各业所带来的发展机遇。全书图文并茂、讲解细致；内容层次分明，由浅入深，并配有大量3D打印机的操作截图。

本书适合想要了解和掌握3D打印技术的读者阅读；对于期望将3D打印技术应用到实际工作中的读者也有一定的参考价值。

3D打印机的构造

第2章

文/坪井义浩 ytsuboi@gmail.com　Twitter @ytsuboi
译/kaku

上一章我们分析了3D打印机的现状，本章将正式介绍3D打印机。3D打印实质上包含了打印立体物品的各种方法，在此将介绍几种代表性方法。

热熔解积层法

热熔解积层法（Fused Deposition Modeling method，FDM法），是桌面3D打印中最常采用的方法。FDM法将热可塑性树脂（加热后会软化，冷却后会凝固的树脂）加热，并从细喷射管前端重复挤压，通过堆叠累积来塑形。

在五金店等地方，可以买到热熔胶枪（或打胶枪）（图1），热熔胶枪可以将热可塑性树脂装入加热器中，软化后喷出树脂进行粘合。

3D打印机的构造

这一部分，笔者将以Ultimaker[①]为例，对消费级3D打印机的构造和这方面的专业术语进行详细说明。

▼照片1　热熔胶枪



步进电动机

3D打印机未使用普通打印机头，取而代之的是电脑控制的小型热喷射枪，通过电机挤压出喷射枪的棒就可以了。

3D打印机的电动机常被称为步进电动机。所谓步进电动机，就是电动机的轴在电子信号控制下，可以按角度转动。例如常见的200步电动机，因一周可分为200步来完成，所以每一步就旋转1.8度。通过简单的回路控制，就可以按比例精确定位，所以这种装置的使用非常频繁。

步进电动机旋转运动与直线运动之间的转换方式也是固定的。用长螺丝钉连接步进电动机的轴并旋转，就可以保持螺丝帽不动，从而改变螺丝帽在螺丝钉上面的位置（图1）。这种方法被称为推送螺丝构造，以前曾被应用在软盘驱动器的针头移动上。

除此以外的方法，还有传送带驱动。转动步进电动机上的同步滑轮，带动同步带（齿形带），

① http://www.ultimaker.com

从而进行直线移动（照片2）。家用喷墨打印机使用的也是传送带驱动方式。

纤维

原料选用直径1.75mm或3mm细棒状纤维树脂。纤维树脂也分为几类，常用的有ABS（Acrylonitrile Butadiene Styrene，丙烯腈－丁二烯－苯乙烯共聚物）树脂和PLA（PloyLactic Acid，聚乳酸）树脂（第3章中有详细介绍）。ABS树脂是工程塑料的一种，强度高、耐热性好。PLA是聚乳酸，被称为可降解塑料或生物塑料树脂。与ABS相比，PLA的纤维能在低温下熔解，且熔解时可粘合。图2为可以买到的3mm PLA卷盘，纤维的颜色也很丰富。

挤压机

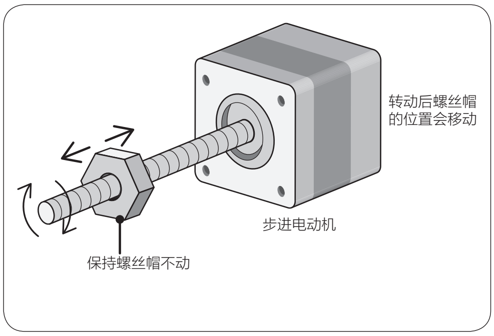
将纤维的前端，插入打印机的挤压机中（照片3，打印机的“头部”）的纤维插口中，打印时，随着电动机齿轮的转动，纤维就会被挤压向加热器和喷头。纤维被加热器加热，变软后可以从细喷头挤出到桌面上。

Ultimaker采用的是纤维输送装置与热端分离的Borden方式，但RepRap或Replicaor等机器中一体化的设计更为常见。图3为挤压机和输送装置一体化的机型。

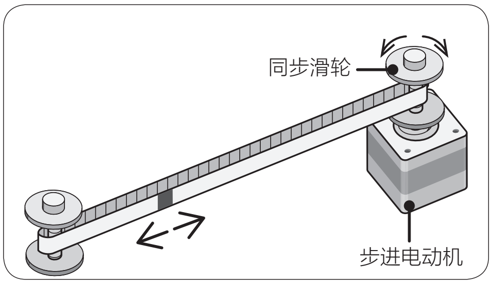
纤维的温度管理

3D打印机的热熔胶枪及其温度均由电脑控

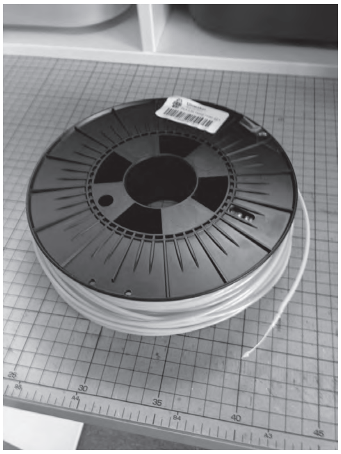
▼图1　推送螺丝构造



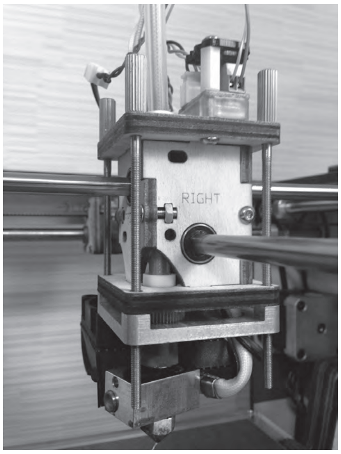
▼图2　传送带驱动



▼照片2　纤维盘



▼照片3　挤压机



制，这与普通的热熔胶枪有所区别。五金店中所售热熔胶枪的加热器没有温度控制，所以在长时间通电的情况下，容易因过热致使树脂垂直从喷头中流出。3D打印机若发生此类情况，就不能正常地输出打印，所以需要由电脑将其控制在合适温度范围。比较麻烦的是，ABS的纤维温度是不固定的，需要调控出精准的温度，以避免纤维从喷头垂直流出或堵塞喷头，且满足熔化塑形的要求。

喷头和积层精度

喷头前端口的大小（喷头直径）有0.5mm和0.35mm两种类型。Ultimaker喷头直径为0.4mm。喷头直径和积层精度并不一致。例如，直径0.35mm的喷头很有可能是0.3mm的积层精度。但是，直径0.5mm的喷头，若设置输出树脂积层精度为0.25mm的话，X轴、Y轴方向的解析度就会降低。大直径喷头的制造物虽然坚固，但精细度方面不及小直径喷头。

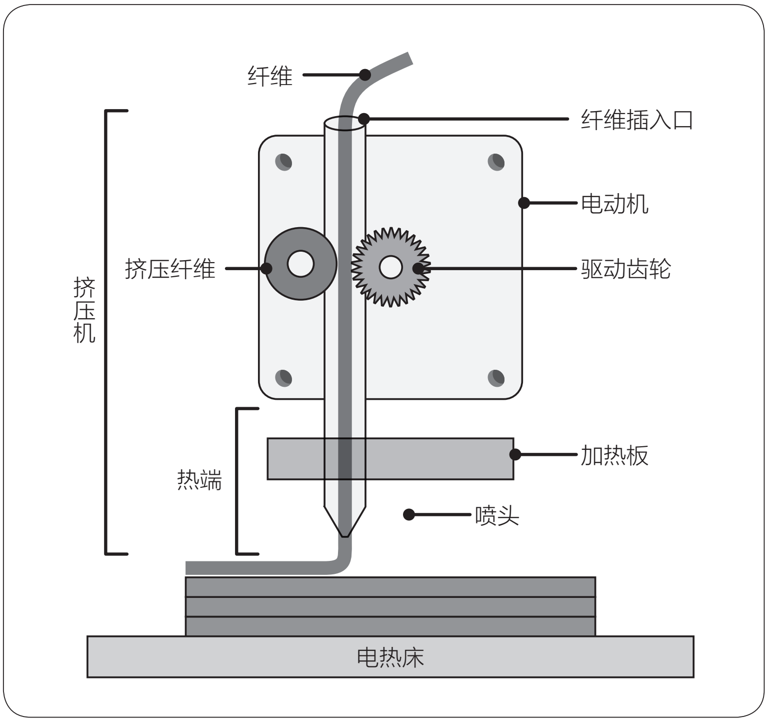
积层方式的轴

采用积层方式的3D打印机，在X、Y两轴构成的平面上逐层打印树脂纤维，整体则按Z轴方向堆叠（图4）。不同3D打印机的打印方向都各具特色，研究一下还是很有趣的。Ultimaker的X轴、Y轴的进料头由传送带和滑轮带动，Z轴上，通过螺丝传送结构进行垂直运动（照片4）。

加热床

加热床是可以通过加热器加热的打印托盘。如果是普通的打印托盘，输出后的树脂冷却收

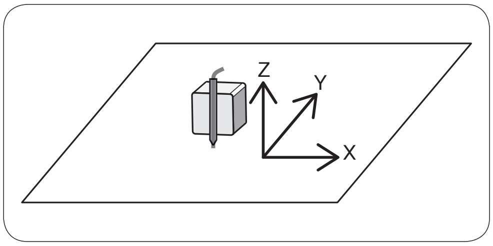
▼图3　挤压机的构造



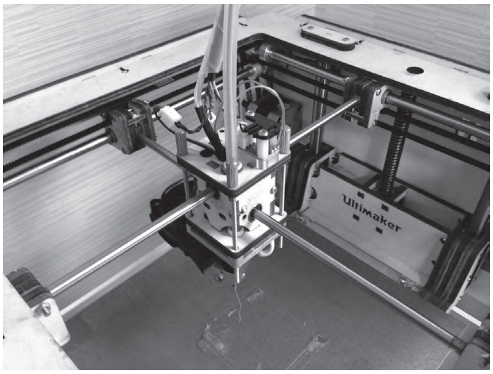
缩后，会从托盘剥落。特别ABS材料冷却后收缩更加明显。为了避免这种情况，就要通过加热器对打印托盘进行加热。

Ultimaker的打印托盘就没有附带加热器。如照片5所示，树脂板材质的打印托盘上，贴有蓝色的保护垫。即使是使用冷却后收缩较少的PLA，输出后很容易贴合在打印托盘上，此

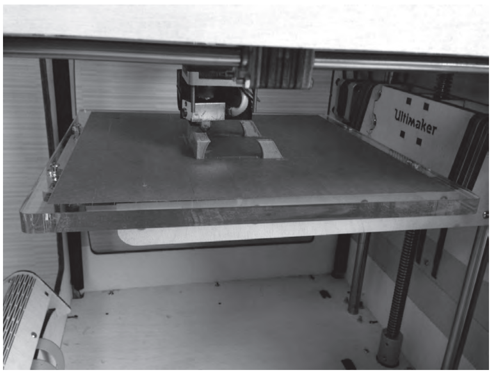
▼图4　积层方式的X、Y、Z轴



▼照片4　X、Y、Z轴



▼照片5　打印台



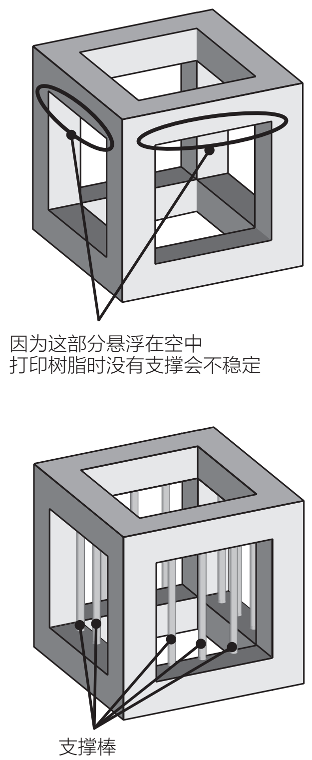
类托盘上，若不使用保护垫，取下制品时容易损伤打印托盘。

支撑棒

若想打印如图5中形状的物品该怎么办呢？此类物品中，无支撑的部分一定要用树脂打印，否则可能会下垂。然而，实际制作中多少都会下垂，所以，若想完美塑形，就需要有支撑棒。支撑棒应便于取下，否则打印后会造成困扰。若只有一个挤压机，可以使用与打印材质相同的纤维制作支撑棒。打印完成后，再将其切除。一般情况下，支撑棒都要制作成易于切除的形状。

若有多个挤压机，就有更简单的方法。可以使用易去除的材料纤维（例如水溶性材料）打印支撑部分。常用的支撑材料有PVA（polyvinyl

▼图5　很难进行3D打印的形状



alcohol）等。例如Makerbot公司的Replicator有两个挤压机，使用其官方售卖的PVA纤维制作支撑棒的话，打印结束后只要用水浸泡就可以去除。

◆ ◆ ◆

热熔解积层法，是美国Stratasys公司的创始人S.Scott Crump先生于1988年发明的专利。Stratasys公司是世界上最大的3D打印机提供商，去年年末，被以色列喷墨3D打印机制造商Object公司收购。

消费级3D打印机大部分都采用热熔解积层法。开源的RepRap如此，Makerbot公司的产品也是这种方式。另外，3D Systems公司也发售的Cube及Cube X也是使用热熔解积层法的3D打印机。热熔解积层法3D打印机的一大特点是，使用ABS树脂输出时可以保温。但是，因为Stratasys公司的美国专利[②]现在仍然有效，所以其他消费级3D打印机没保温功能。

喷墨法

喷墨法（图6），刚才所说的Object公司就是采用这种方式。跟传统的纸印刷喷墨打印机一样，从喷头喷出材料的细小粒子或黏着剂，通过重复累积来塑形。Object公司的3D打印机，使用的材料为紫外线照射下会固化的UV硬化性树脂（用紫外线固定的树脂）。

喷墨法与喷墨打印机一样，优点在于可以喷出同样细小的颗粒，与热熔解积层法相比，16μm（0.016mm）的积层精度可以进行更精细的塑形。另外，Object公司的3D打印机搭载了多个喷头，可以分别喷出不同的树脂。例如打印车轮时，这个功能可以实现使用不同树脂材料打印车轮和轮胎，轮胎还可以专门用弹力与橡胶相当的树脂来打印。另外，也可以在透明树脂中加入有色树脂进行打印，在产品展示会的展品中就有这样制作出的人体模型。

粉末积层法

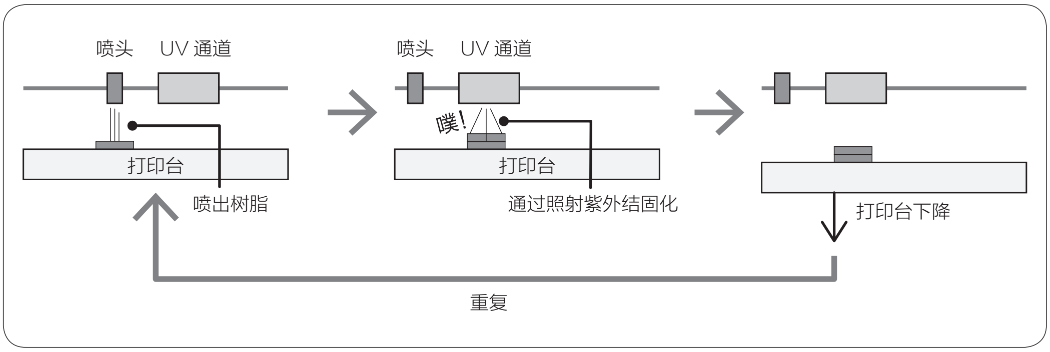
粉末积层法大致可以分为两类。

粉末固定积层法

第一类是粉末固定积层法，先铺上用石膏或树脂等材料做成的粉末，然后从上方喷出粘

② High temperature modeling apparatus（USPTO#6,722,872）

▼图6　喷墨法



着剂重复印刷断面来对物体塑形（图7）。此类3D打印机中，比较有名的是美国Z Corporation的Z打印机，Z打印机还具备着色功能，所以能打印出带颜色的3D物体。

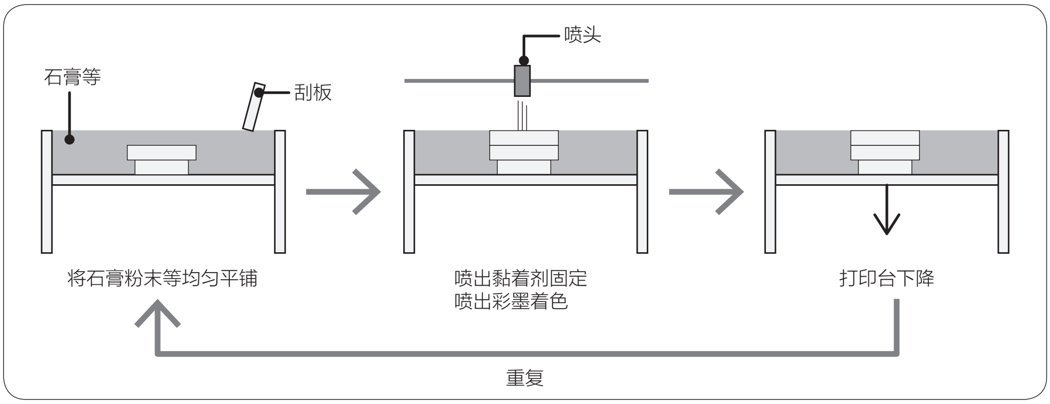
粉末固定积层法，最终是将埋在粉末中的物体挖出来，所以不需要支撑棒。支撑棒不能用于物体塑形，所以属于浪费材料费的额外成本，粉末固定积层法可以完美解决这个问题。但是，这种塑形方法下，材料的粒度并不是很细致，打印效果稍显粗糙，颗粒感较强。

粉末烧结积层法

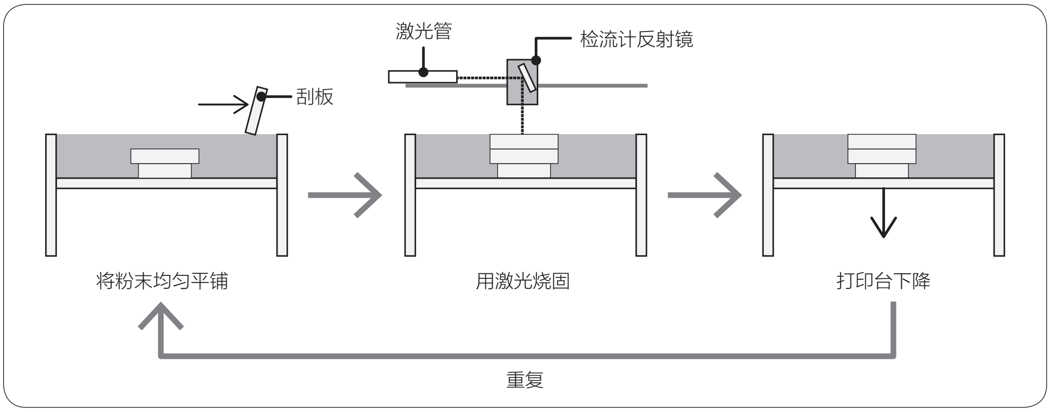
另一类被称为粉末烧结积层法。这种方法，采用激光烧固金属或树脂的粉末，代替石膏粉来固定积层（图8）。要使激光照射到任意地方，一般要使用检流计式扫描仪。检流计式扫描仪，由多个通过前置反射镜来改变反射角度的检流计式电动机组成。这种情况只对平面（X轴、Y轴）方向处理较好，所以需要两台电动机。

粉末烧结积层法与其他方法不同，可以使用金属粉末，金属3D打印也成为其一大特点。采用这种方法的厂商，还有德国的EOS公司和美国的3D Systems公司。另外，3D Systems公司在去年年初已经被Z Corp.公司收购了。

▼图7　粉末固定积层法



▼图8　粉末烧结积层法



光固化成型法

光固化成型法（Stereolithography，可以省略为SL或SLA等），是用紫外线灯照射液体状态的光固化树脂，从而固化塑形的方法。这个方法可以说是3D打印的始祖。光固化成型法打印品质精细，但材料只限于光固化树脂。

▼图9　检流计式扫描仪

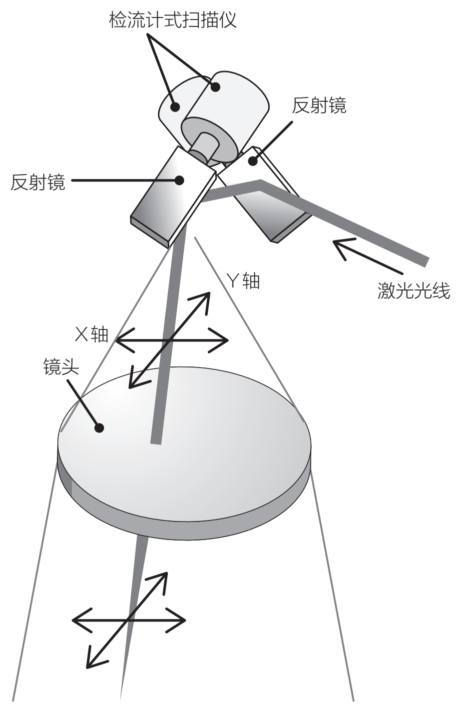


图10是从树脂层（用树脂做成的台子）逐步打印输出的示例。与此相反，从上往下利用激光照射使打印物品逐渐沉淀的方式也可以。另外，光固化成型法为了在任何情况下都可以使用激光，常会用到检流式扫描仪。

最初提出光固化成型法专利的是日本人，1980年名古屋市工业研究所的小玉秀男申请了专利。现在，小玉先生是位律师，没有进行自身审查请求，在审查请求期限过后，由3D Systems的Charles W. Hull取得了专利。

去年，Formalabs的消费级光固化成型法3D打印机Form 1备受关注。Form 1打印机采用激光固化光固化树脂，从而进行积层塑形。Form 1是通过Kickstarter群众募资（由普通人募集而来的资金）网站募集资金，原本计划1个月募集10万美元，结果不到3小时就成功达成目标，因而受到关注。最终Formlabs在30天中成功达成了294万美元左右的集资。

之后，Form 1因侵犯专利诉讼而再次引发关注。控告Formlabs和Kickstarter的是3D Systems。3D Systems声称Form 1侵犯了该公司的美国专利[3]请求项1和23。虽然直到去年11月审判还没

③ Simulaneous multiple layer curing in stereolithography（USPTO #5,597,520）

Column 检流计式扫描仪

这部分内容是题外话，但检流计式扫描仪的应用确实十分广泛。例如，在墙壁或烟雾中用激光成像的激光探照灯，在这种装置中加入检流计式扫描仪，通过移动激光位置来形成影像。激光虽是点状，但是利用人类眼睛的残像效果就能看到最终成像。

另外，半导体电容上，经常可以看到型号等细小文字，这也是使用安装了检流计式扫描仪的激光加工机来镌刻的。

出结果，但此次诉讼关系到消费级3D打印机发展的最大障碍——老牌3D打印机提供商所持有的专利问题，因而很多人都在关心此事件的走向。

另外，除了激光固化塑形的方法，还有采用聚光灯固化树脂的打印机B9Creator。B9Creator也是Kickstarter通过融资制造的3D打印机，使用了一般办公室或家庭常见的1024×768聚光灯的光线照射。因为聚光灯不能高速开关，所以树脂池下面的快门设计也是非常有意思。B9Creator当初的融资目标仅为5万美元，现在实际的融资已达到51万美元。

层片积层法

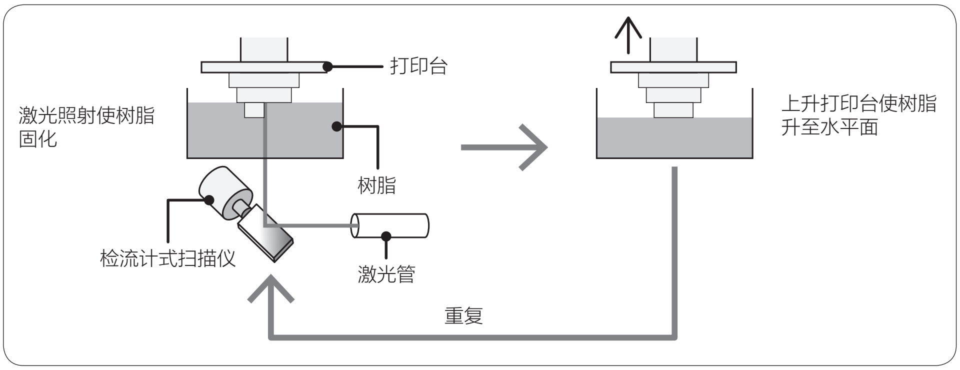
虽然笔者还未见到过采用此种方法的3D打印机，但层片积层法也是很有意思的一种方式。这种方法将纸或者树脂的层片粘合来制造立体物品。用喷墨打印机等在想要制造的物品的断面刷上颜色，然后将层片按照断面的轮廓切好。用黏着剂粘合切好后的层片，从而制作立体物。

Mcor Technologies公司的Mcor IRIS 3D color printer采用了这种方法，这个打印机中内置的喷墨打印机是爱普生的上市产品。

◆ ◆ ◆

到此为止，我们介绍了各种各样的3D打印方法，个人3D打印机大部分采用的都是热熔解积层法。近年来，采用立体光固化成型法的打印机也陆续出现，吸引了很多目光，但是在较长的一段时间内，热熔解积层法的打印机仍然会是主流。

▼图10　光固化成型法



第3章 尝试制作3D模型

文/山田齐　工房Emerget　Http://www.emergeplus.jp Twitter @emergeplus
译/陶旭

3D打印首先需要构建3D模型数据。当然，我们可以下载网上的3D数据直接使用，但是自己来构建原创3D模型，则更能享受到3D打印的乐趣。因此，笔者将在本章介绍几款免费的3D-CAD软件，并演示如何使用3D-CAD软件构建iPhone5机座模型。

认识3D-CAD

大多数人可能还不太熟悉3D-CAD软件（以下简称3D-CAD），但对于最近颇受媒体热捧的3D打印机来说，3D-CAD是必不可少的。也就是说，必须要先有3D模型数据才能进行3D打印。

所谓3D-CAD就是可以在电脑上自由构建三维模型的软件。例如我们使用电脑制作及打印立体纸模型的平展纸样时，会使用Adobe Illustrator等制图软件或2D-CAD软件，像在纸平面上画线那样在电脑中绘制图形，然后打印出来。如果把这个过程扩展到三维范畴、即在电脑上构建立体模型，那么使用的软件就是3D-CAD。

免费的3D-CAD

最近颇为流行廉价的3D打印机，而支持这种打印机的3D数据格式几乎都是STL（Standard Triangulated Language），市面上也出现了不少功能强大且可以输出STL格式文件的免费软件。笔者挑选了几款可以支持多平台且非常适合新手入门的软件介绍给大家，其中对应于Linux系统的软件则介绍其浏览器版本。

而且安卓版和浏览器版的软件都具备在网上共享3D模型的功能，所以我们可以直接使用别人在网上共享的3D模型，或是以此为基础进行修改。

安卓版

• Autodesk 123D Design ①

这是由3D-CAD的鼻祖Autodesk公司提供的免费3D-CAD。这款软件虽然刚发布不久，但提供了可以将基本3D组件组合起来的功能和利用二维图形制作3D组件等功能，并具有灵活、直观且容易使用的UI。该软件不仅有Windows、Mac版本，还有支持iPad的简易版本，而且虽然不能导入STL文件但可以导出STL文件。遗憾的是现在还没有日语版，只能使用英语版②。

① http://www.123dapp.com/design
② 目前已经支持中文版。——译者注

• SketchUp Make [3]

这款软件的原型是Google从美国Last Software公司收购的建筑用3D-CAD，原名为Google SketchUp。现在又被Trimble公司收购，目前正在进一步开发中。在操作方法方面，该软件的UI在二维画面上实现了类似用铅笔在纸上画草图的效果，通过其独特的设计解决了3D设计中让人颇感棘手的“三维绘图”这一建模难题。这款软件的原名是SketchUp，在今年（2013年）5月从SketchUp8升级到SketchUp2013时改名为SketchUp Make（笔者认为这是因为考虑到3D打印机等Maker市场正在逐渐扩大）。该软件有日语版[4]。虽然其标准设计不支持STL，但是安装了由Trimble公司的扩展仓库（Extention Warehouse）提供的STL插件后就可以输出STL[5]了。

浏览器版

• Autodesk 123D Design [6]

浏览器也可以使用这款123D Design。这个版本虽然没有安卓版的功能强大，但也具备了制作简单作品的各项功能。

• TinkerCAD [7]

正如它的名字[8]一样，这款浏览器版3D-CAD所使用的UI操作十分简单，即便小孩也能简单操作。其特点是，将一些基本的立体组件像搭积木一样组合起来，设计过程非常简练。即便这款软件如此简单易用，但只要足够用心，也可以完成非常复杂的设计。事实上在今年（2013年）3月末该软件曾被宣告停止运营，但5月份被Autodesk公司收购后，又重新开始了稳定运营。

③ http://www.sketchup.com/products/sketchup-make
④ 该软件支持简体中文版和繁体中文版。——译者注
⑤ http//:extensions.sketchup.com/en/content/sketchup-stl
⑥ http//:apps.123dapp.com/design/
⑦ http//:tinkercad.com/
⑧ Tinker意为粗笨工匠、顽皮孩子。

使用TinkerCAD体验3D建模

下面我们就试着使用TinkerCAD来一起体验3D建模吧。TinkerCAD的UI非常简练，可以通过搭积木的形式来完成建模，因此即使是初次接触3D-CAD的用户也能很快掌握基本操作。

TinkerCAD的用法

运行环境

TinkerCAD是浏览器应用程序，不需要安装到电脑上，只要有支持WebGL的浏览器即可。在现有条件下推荐如下配置。

- 操作系统：Windows Vista以上、Mac OS X 10.6以上。
- 浏览器：Chrome10以上、Firefox4以上。

本文示例中使用MacBook Air、Mac OS X Mountain Lion（10.8.3）里的Safari（6.0.4）来运行软件。虽然Safari不是推荐使用的浏览器，但可以通过如下操作来应对：先从[偏好设置]→[高级]中选择[显示开发菜单]，然后到开发菜单选择[开启WebGL][9]。

⑨ 笔者通常是使用Chrome，但因为鼠标手势插件妨碍鼠标操作，无奈之下只能用Safari。如果出现无法正常操作的情况把鼠标手势设为无效即可。

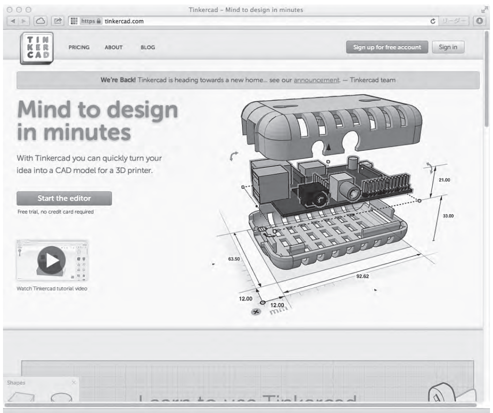
注册账号

在使用TinkerCAD时，首先在浏览器中打开https://tinkercad.com/（图1），注册免费帐户，流程与其他网络服务的注册流程基本相同。先点击画面右上方的Sign up for free account，需要填写的内容包括姓名、邮箱地址、密码、出生年月日（图2）。

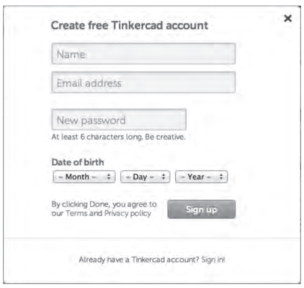
体验自学教程

完成注册后首先会进入在线教程（图3）。这里准备了很丰富的教程，如果都学一遍应该就会对TinkerCAD的操作比较熟悉了。每个教程都不会耗时太长，所以如果用户想尽早掌握软件的操作，笔者非常推荐这些教程。

▼图1　开始界面



▼图2　帐户注册界面

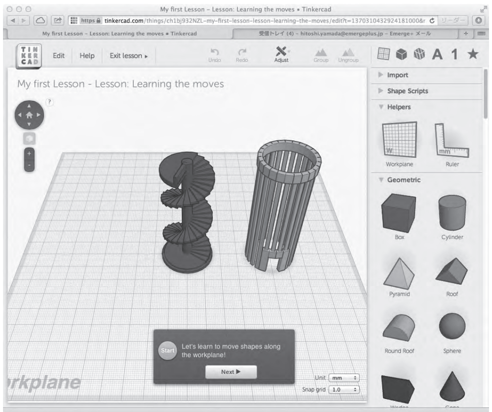


操作面板（Dashboard管理画面）

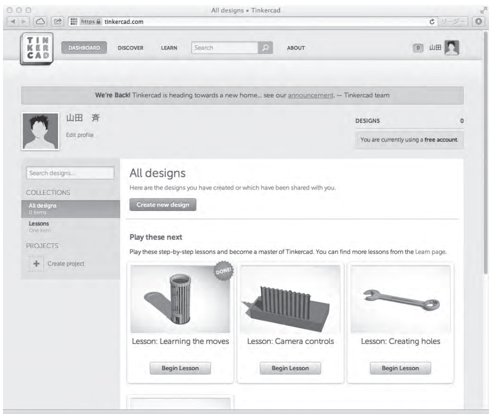
我们在体验过一个教程后就会进入操作面板[⑩]（图4）。在这个操作面板上会显示自己之前制作的模型及教程的缩略图，把鼠标放在缩略图上，点击出现的齿轮图标，就可以对这些模型自由进行添加、修改、删除、公开/不公开、重命名等操作。此外在操作面板上还可以修改个人资料。在面板上方有各种菜单，从这里可以访问其他用户的资料以及观看视频演示等，本文不再具体介绍。

⑩ 仅首次使用时会进入教程。

▼图3　在线教程的界面



▼图4　操作面板



建模画面

点击操作面板上的Create new design按键进入设计面板（图5）。如果要修改之前的数据，则把鼠标放在相应的缩略图上，点击出现的Tinker this按键。

关于菜单

为了加深对建模操作的理解，我们来大致浏览一下整个设计面板。设计面板由菜单栏、侧边栏和主窗口三大部分组成，先来看一下最上方的菜单栏（图6）。

最左侧的Design相当于通常应用程序中的“文件”菜单，用于设计的新建、保存、修改等设置。想要用3D打印机打印时，在Download for 3D Printing下点击STL按键（图7）。点击Order 3D Print，可以向提供3D打印云服务的Ponoko、Shapeways等网站直接下单。点击Close则可返回操作面板。TinkerCAD会自动保存数据，这点十分方便。

Edit、Help、Undo、Redo则与我们所熟悉的用法相同，这里不再多说。

Adjust菜单中有多组件排版Align和镜面映射Mirror。

Group是非常重要的菜单项，不仅可以将多个组件合并，而且也可以对某个组件进行切削或开孔（图8-10）。也就是说，这个功能实现了3D建模的“或”和“非”的逻辑运算。Ungroup则是将其逆向分解、还原。

最右侧的5个图标是侧边栏制图的快捷方式，可以分别点一下试试。

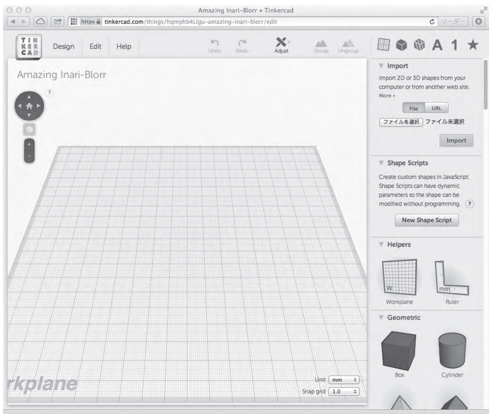
关于菜单

接下来介绍侧边栏。

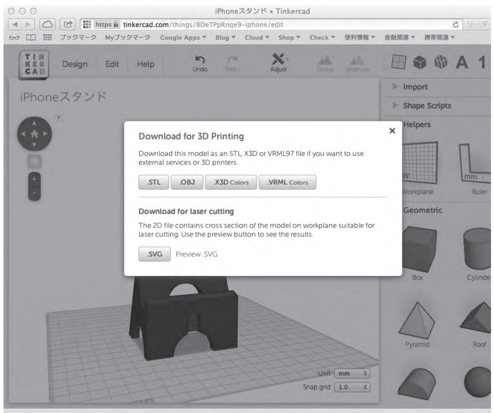
Import菜单

最上方的Import用来导入其他模型（不限于STL格式），既可以用来导入本地文件，也可以导入网上的模型。

▼ 图5　设计面板



▼ 图7　STL下载界面



▼ 图6　菜单栏



Shape Script菜单

下面这个Shape Scripts是TinkerCAD建模的扩展功能，Autodesk公司收购ThinkerCAD后将这部分功能开放给了免费用户。这是使用JavaScript自定义/注册图形并将其应用于建模的高效工具。关于这部分本文中不再截屏解说，读者亲自去试一下，才会发现这个功能的有趣之处。

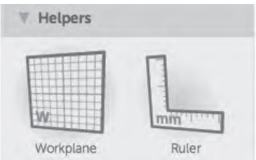
Helpers菜单

Helpers是在3D建模中非常重要的工具（图11），因此在这里稍加解说。

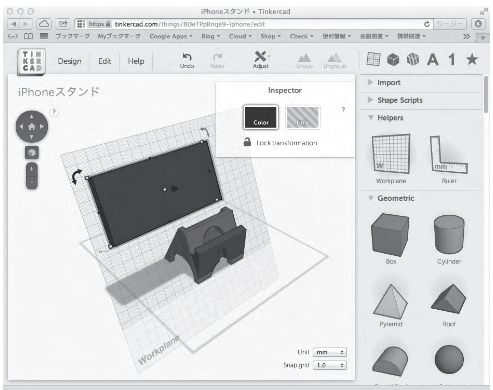
• Workplane

Workplane是可以简单地在二维平面上构建三维模型的有效工具，在实际操作中会经常

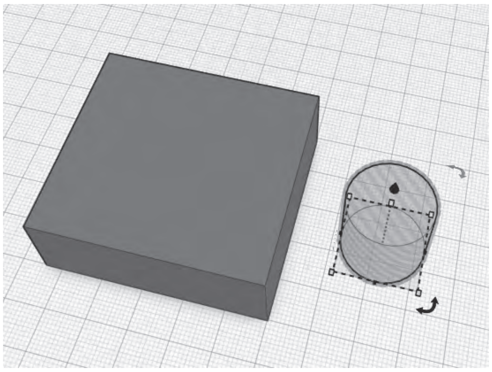
▼ 图11　Helpers功能



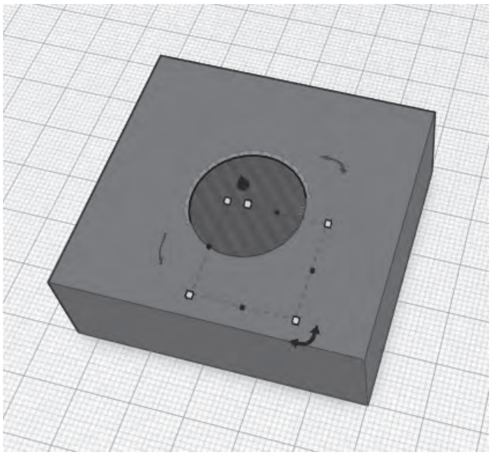
▼ 图12　在组件斜面上使用Workplane



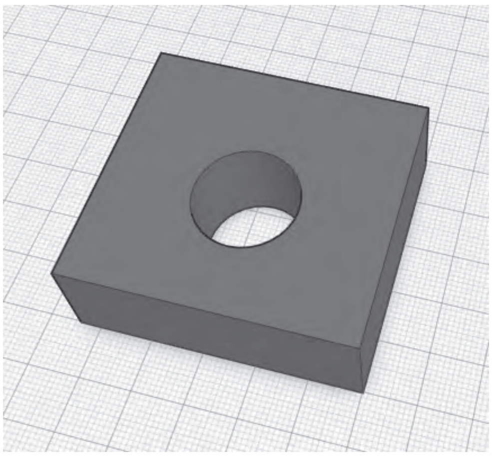
▼ 图8　Box（立方体）和用于开孔的Cylinder（圆柱）



▼ 图9　在准备开孔的地方放置Cylinder



▼ 图10　使用Group开孔



使用到(图12)。以模型的某个面为基准设定Workplane平面，并在此平面上固定3D组件，这样就可以不用担心组件倾斜，而可以直接像搭积木一样实现组件的叠加和组合[⑪]。

另外由于该软件具备以mm/inch为单位的网格对齐功能，故而可以简单地借助鼠标来调整一定的位置。可在主窗口右下方的Unit中选择mm或inch来确定对齐的单位，在Snap grid选择对齐的具体数值。

实际使用时只需在点击Helpers中Workplane的缩略图后，再点击主窗口中任意组件的任意面粘贴即可，这样就会只在该面的无限面上对组件进行网格对齐，也不会涉及其他组件。

在空白的地方放置Workplane就可以将其恢复到默认状态。

• Ruler

这也是一个功能强大的建模辅助工具，可以暂时自由返回/设定原点。点击Ruler缩略图后，就可以在基准位置配置标尺(图13)。

例如利用这个工具，就能够以某组件的一个角为基准，实时确认某个具体坐标上是否放置了其他组件。Ruler更方便的功能是，在放置组件后可以点击各尺寸并修改相应数值，从而正确地指定该组件的相对位置和组件尺寸。点击“ ×”图标可以去掉标尺。

仅使用鼠标进行操作的时候，有可能会出现“看起来各个组件似乎都连在一起，但实际上却相互分散”的情况，Ruler则可以有效避免这样的设计失误，因此会经常用到它。实际上Ruler如果能同时显示及设定角度，那它的功能将更加强大，这样的改进很值得期待。

Geometric~Extras菜单

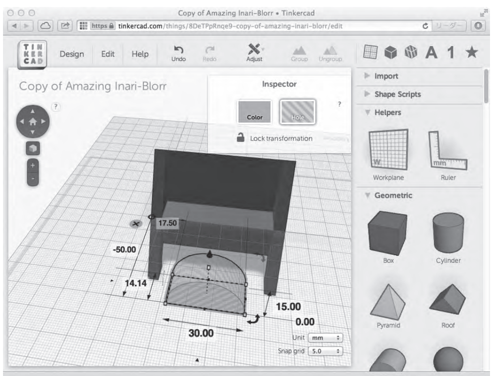
这里包含了TinkerCAD标准版提供的基本组件。大多数建模都可以通过这些组件的自由变形和组合来完成(图14、图15)，并且还有很多为小朋友用户准备的可爱素材。

主窗口

最后再来看一下建模的主窗口。这个区域

⑪ 不过，利用组件的Z轴移动柄可以使组件相对于Workplane面进行偏移摆放。

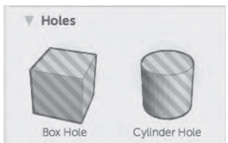
▼ 图13　将Ruler的基准运用于上方的组件



▼ 图14　Geometric



▼ 图15　Holes



的基本操作是，先从侧边栏选择图形项目，然后在Workplane上拖动鼠标以确定位置，并使用各手柄进行变形/转动/移动等操作（图16）。

相机视图

在主窗口右上方有修改模型名称和视角的图标（图17）。模型名称最初是随机命名的，用户可以在操作面板上对其进行修改。

相机视图是3D建模不可缺少的功能，我们自上而下逐个讲解这些图标。

点击最上方的房子图标，会默认显示模型整体。点击箭头图标，可以从前后上下左右的各个视角观察模型。选择某一组件后点击正中间的图标，可以整体显示该组件。而最下面的“+”“-”符号分别是我们所熟悉的放大和缩小功能。另外点击右上的“?”则会出现鼠标操作的快捷方式（图18）。

虽然并没有可以上下左右移动画面整体的Pan功能的图标，但可通过按Shift键+鼠标右键来完成该操作。这个快捷方式非常方便，用户最好能记住。

设定网格

正如上文Workplane部分介绍的那样，在主窗口的右下方有设定网格对齐的菜单项。用户可以根据情况进行选择，非常方便。

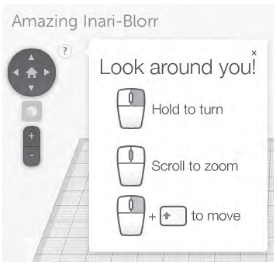
监控器（颜色/开孔/变形锁定）

选择了组件之后，右上方会出现修改组件属性的对话框（图19）。用户可以选择喜欢的颜色，设定开孔属性，以及开启锁定功能，避免因疏忽导致组件变形。

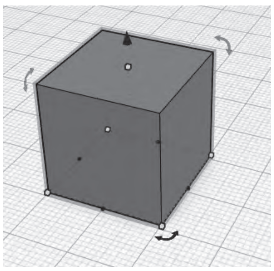
尝试iPhone5机座建模

下面就来实际构建3D模型。我们使用3D打印机的理由之一是“想要亲手制作自己喜欢的小物件”。笔者想要一个不用把iPhone5拿在手上也可以用它来看电影的小工具，想来这也

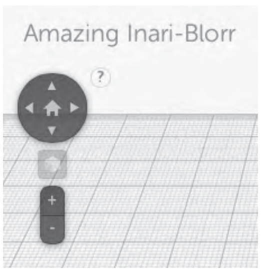
▼ 图18　相机视图功能的鼠标快捷方式



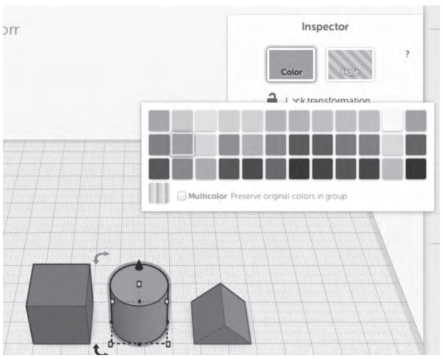
▼ 图16　用于组件变形/转动/移动的各手柄



▼ 图17　模型名称和相机视图图标



▼ 图19 组件修改对话框



不是很难的东西，应该可以用TinkerCAD做出来，所以就尝试制作了一个iPhone机座。

首先来确定iPhone5的相关尺寸

要想给iPhone做机座，就要先知道它的相关尺寸，在Apple的网站上有它的尺寸数据[12]。

- 高度：123.8 mm
- 宽度：58.6 mm
- 厚度：7.6 mm

思考要做成什么样的(用法/形状)

那么，要做成什么样的机座呢？我们首先需要确定基本设计，这次笔者设想的iPhone机座设计如下。

iPhone机座样式

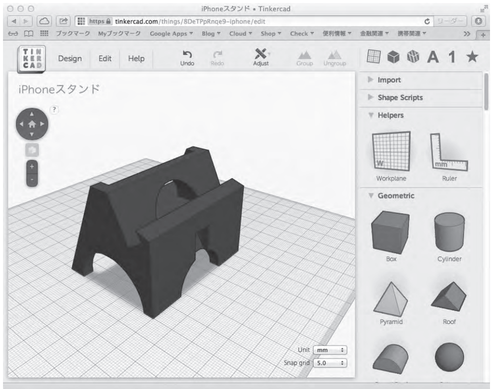
- 能把iPhone横过来放并且角度适合观看电影。
- 最好也能竖起来放。
- 可以插Lightning接口。
- 能简单放住即可(不想用螺丝之类的去固定)。
- 大小要适中，放在桌子上不碍事(最多是iPhone的宽度+α)。
- TinkerCAD和3D打印可以实现的简约造型。

完成效果如图20~图22。

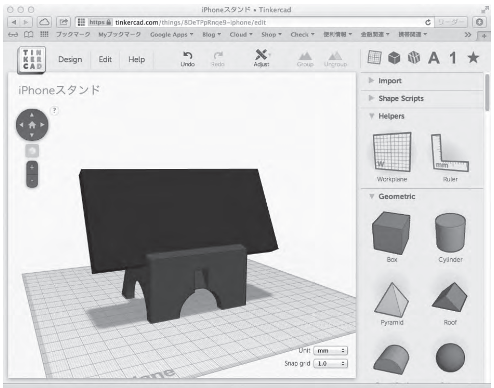
要想制作易用的物品就一定要充分考虑其功能性，那么iPhone机座最重要的是什么呢？首先至少可以让iPhone立起来。

因为不想用螺丝一类的把iPhone固定住，所以为了放置时iPhone不会向后翻，需要在iPhone

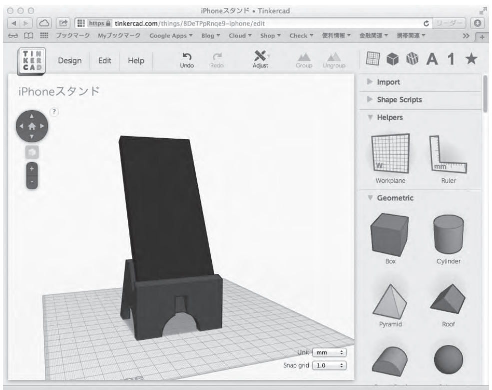
▼ 图20　iPhone机座最终效果



▼ 图21　横向放置iPhone的效果



▼ 图22　纵向放置iPhone的效果



⑫ http://www.apple.com/jp/iphone/specs.html

下方制作挡板。为防止整个机座都翻倒，还必须将机座的后脚撑充分向后延展，直至相对远离iPhone重心的位置。

其次为了保证能正常观看画面，机座不能挡住画面也不能影响触摸屏操作，所以要尽量缩小挡板部分。

为了可以纵向放置iPhone，还需要设置Lightning接口走线的部分。

掌握积层方式3D打印机的特点

大部分人在刚开始构建3D模型时，都是随心所欲地去设计模型，但多多少少经历了实物打印的失败以后，便学会了在建模阶段就有所保留地进行设计。笔者虽然接触积层方式3D打印机只有短短几个月的时间，但也积累了一些类似"这种情况需要让步"的经验，下面介绍一下这些内容。

积层方式打印的数据制作方法

特别要注意尺寸和悬空（后述）的问题。

尺寸

积层方式的特点在于，制作时需要把加热融化的树脂从极细的喷嘴中，如同细丝一般挤出并同时按压。在此过程中，只有细丝的尺寸会出现误差。本次使用的Ultimaker打印机的喷嘴直径为0.4 mm，实际测量的结果也是如此，因此拉出的细丝应该也是0.4 mm，不过直径上大约会存在0.2 mm左右的误差。例如要设计直径为10 mm的轴与孔的组合，如果直接以10 mm直径打印的话，则可能成品的轴直径是10.2 mm、孔直径是9.8 mm，那么轴将无法穿过孔，所以设计时就必须预先考虑将轴直径缩小或者将孔直径扩大[13]。

悬空

这是积层方式3D打印机最大的问题。由于无法使树脂在空中成型，所以有时候需要预先在打印台上用废弃材料做出支撑，再以此为过渡制作出抬升（悬空）部分。当然，如果是设计上有悬空的要求也必须这样处理。

但是使用支撑材料也有缺点。首先是浪费材料，提高了材料成本。其次，如果制作开口向下的深洞，支撑材料很难去除，所以在设计时也要充分考虑完成后如何去除支撑材料的问题。

避免悬空的办法之一是制作模型时选用将模型横向摆放的方式，以此来尽量避免使用支撑材料。

降低打印时间（和材料成本）的数据制作方法

因为打印时间是与挤出树脂的体积大小成正比的，因此减少体积=有效减少打印时间和树脂使用量。简单来说，有两个办法。其一是在3D打印输出时通过设定切片机应用程序来尽可能节省树脂[14]。其二是在设计时采用开口或内凹等方法省去不需要的部分。笔者制作的iPhone机座也做了这方面的尝试。

充分考虑材料特性的数据制作方法

积层方式3D打印材料中最常见的是ABS和PLA。

⑬ 即便实物直径刚好都是10mm，轴和孔也会因为组合在一起时过紧而容易损坏。如果想要轴比较流畅地转动，则要预留出一定的空间，这在机械工学方面称为吻合。当然也可以在打印好以后使用钻等工具扩孔。

⑭ 下一章中会具体说明。

ABS（Acrylonitrile Butadiene Styrene，共聚物合成树脂）

ABS是树脂制品中使用最广泛的原料之一，这种工程塑料多用于电脑、家电、汽车等领域。ABS虽然被广泛使用而且可靠性较好，但网上也经常有人抱怨这种材料在用于积层成型时与打印台密着性欠佳。甚至有时在成型过程中从打印台上掉落，弄得乱七八糟。很遗憾笔者没有使用ABS材料的经验，没什么可以吐槽的，不过笔者认为，问题的关键可能在于ABS材料是否与打印台充分接触。

PLA（PolyLactic Acid，聚乳酸）

笔者通常使用的材料是PLA。这种材料是可降解性塑料，被称为环境友好型树脂。这种树脂的性质与ABS正好相反，与打印台附着性良好且容易成型，但完成后不易从打印台上剥离，较薄的东西有时会在剥离的过程中损坏。考虑到打印台面极易受损，所以笔者一般会贴上胶带，但有时保护胶带又会黏在成品上。所以在使用PLA时需要尽量减少与打印台的接触面积，可是如果想减少接触面积就容易出现悬空设计，这一点也很让人矛盾。

实际使用TinkerCAD制作数据

下面我们使用TinkerCAD按照预设的样式来进行设计。这里为了最大限度发挥TinkerCAD的简捷特性，在设计上没有特别纠结细节，采用功能至上的原则。

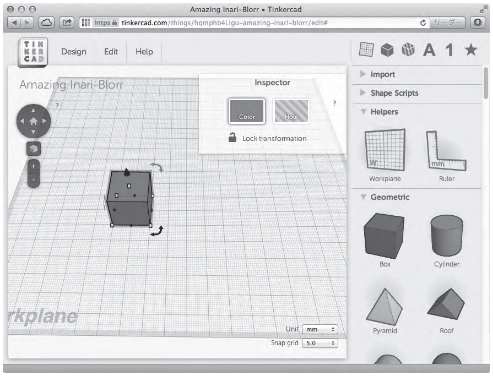
首先把机座最下面的基台部分Box放在适当的位置上（图23）。

其次，操作手柄使将BOX的大小调整为设计尺寸。这里设为宽度65 mm、纵深40 mm和高度35 mm（图24）。每一个都是5的倍数，因此将Snap grid设定为5 mm比较易于调整。各处尺寸的设定依据如下。

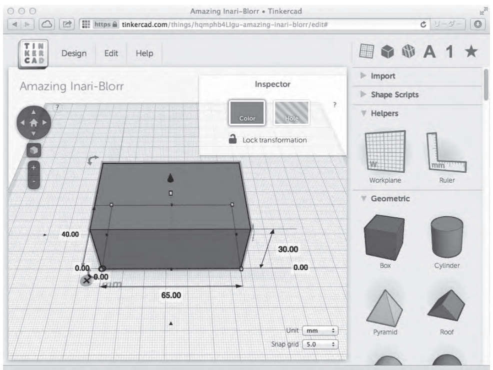
- 宽度65 mm：稍宽于iPhone5纵向放置时的宽度。
- 纵深40 mm：为了便于观看画面将深度按照iPhone5倾斜60度的标准设计。
- 高度35 mm：考虑了纵向放置时Lightning接口数据线的走线空间。

考虑到数据线的走线空间，机座整体设计

▼ 图23　设置基台Box



▼ 图24　调整尺寸



得偏高。这个高度可能会造成机座iPhone在纵向放置时向后翻，为了避免这样的情况就需要充分加长后部撑脚。将Roof转90°，在BOX两侧各配一个宽度为5 mm的撑脚（图25）。撑脚的长度设为14.14 mm（角度为60°），这样可以避免翻倒。认真推算的话应该可以一次性算出准确的重心位置，但也可以反复尝试得出结果。

设置另一侧的撑脚时，使用Edit→Duplicate的功能复制组件并横向平移即可（图26）。

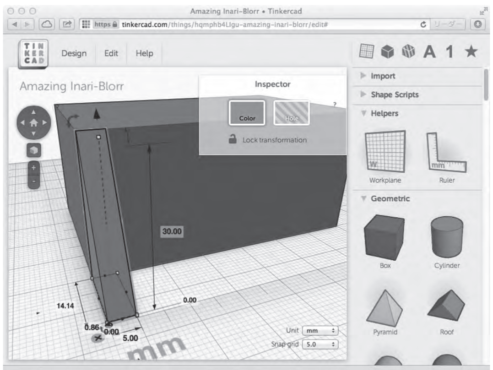
仔细观察的话可以发现BOX和ROOF之间稍有缝隙，这是3D建模中经常出现的情况，如果不注意的话打印时实物就会散架（图27）。这时可以使用Adjust→Align把右边的撑脚与左边的撑脚对齐，点击需要对齐部分的灰色圆点即可（图28）。

然后暂且先将这3个组件合并，点击Group（图29），如果整体变为同一颜色则可以确定完成合并了。当然，需要修改时还可以使用Ungroup解除合并。

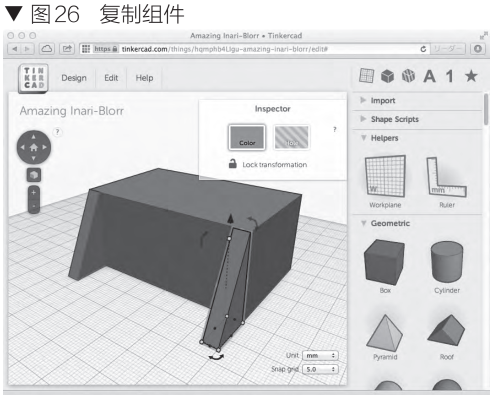
但是现在模型的体积比较大，这样它与打印台接触的面积也会较大，为了减少材料并且减少与打印台的接触，我们需要制造出空隙。使用Round Roof来开孔，将开孔适调整为适当大小，然后在右上方的Inspector中指定Hole。这时就会出现灰色透明的立体图形（图30）。

将其对准基台的侧面推进，选中两个组件并点击Group使其合并后即完成了开孔（图31）。

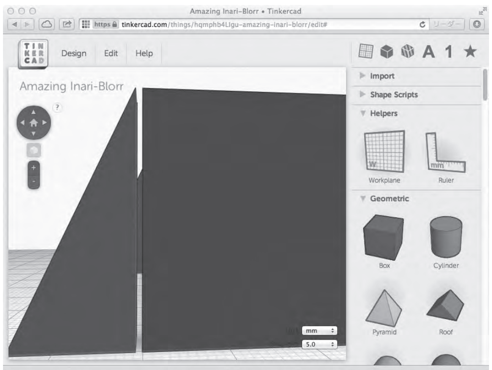
▼ 图25　设置Roof



▼ 图26　复制组件



▼ 图27　出现缝隙



▼ 图28　对齐位置



那么，我们为什么不用Box而用Roof Round来开孔呢？这是为了回避之前提到的悬空问题。如果使用Box的话，基台两侧会出现较长的悬空部分，这可能会造成树脂细丝下垂而无法维持规整的形状，有时还需要另外设计支撑部分。反之如果使用制造的圆弧状空洞，那么悬空的两侧是由宽到窄、逐渐横向延伸的，这样就可以不需要支撑材料了。

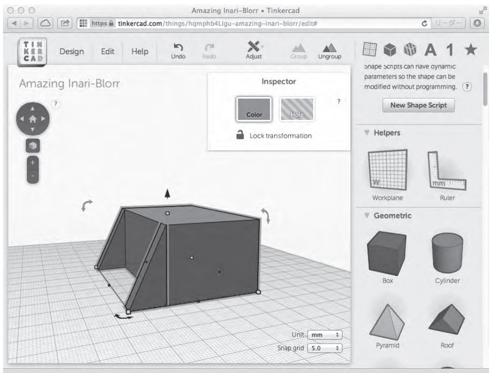
下面构建放置iPhone的部分。将基台的顶面设为Workplane，然后通过组合变形等操作来制作用于支撑iPhone的斜面Wedge。接着，设置与基台撑脚平滑衔接的后部支撑部分（图32），通过Edit→Duplicate的操作，复制相同的形状到另一侧即可（图33）。

下面制作iPhone靠背的斜面。拷贝之前的Wedge，使用Adjust→Mirror将其翻转，即成60°的斜面。如果这个角度不适合观看，之后可以再重新调整（图34）。把这个形状扩展到基台整体的幅宽，这样，靠背斜面就制作完成了（图35）。

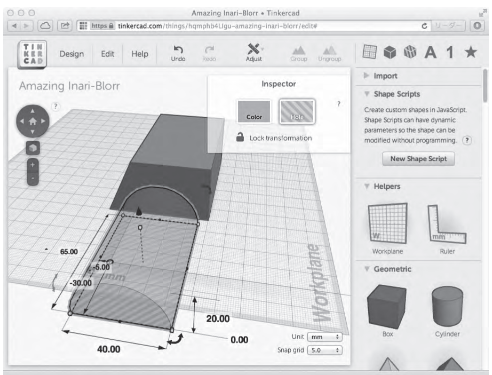
下面制作支撑iPhone的底边斜面。方法也是调整Wedge的形状并使之垂直于靠背斜面（图36）。靠背斜面也需要从中间开孔。图37是合并后的效果。

接下来制作防止iPhone后仰的挡板（图38）。将Workplane设在底边斜面，虽然还是使用Wedge制作挡板，但iPhone5的厚度为7.6 mm，所以为了预留出一定的空间需将底边斜面的幅宽调整为9mm。挡板的高度则设置为横向放置

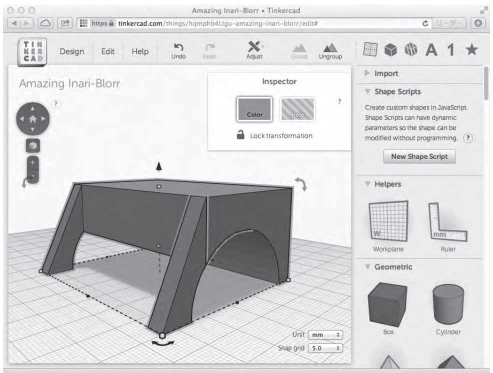
▼ 图29　合并



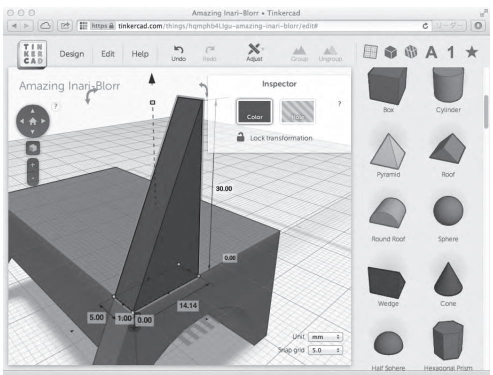
▼ 图30　开孔图形的透明显示



▼ 图31　开孔后的状态



▼ 图32　制作iPhone的支撑部分

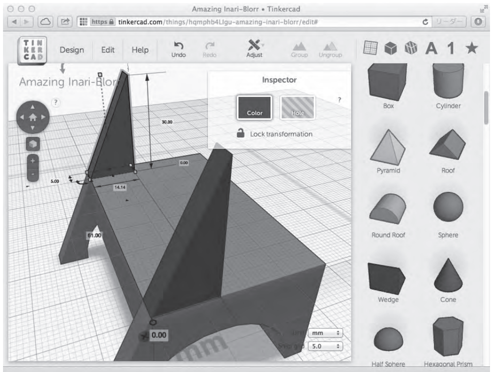


iPhone5时不会遮挡画面的5 mm。

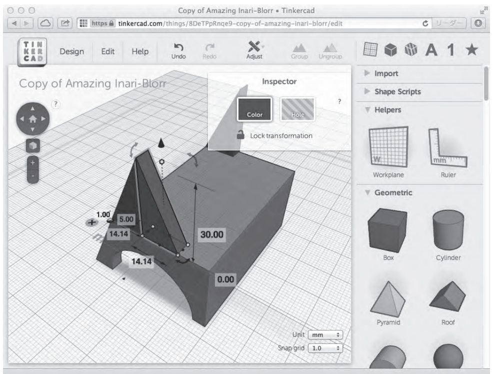
经过这些设计后，机座前面的部分显得有些笨重，再加上考虑到降低成本的原因，所以决定使用Round Roof来削掉一些（图39）。

使用Box的Hole来制作纵向放置时Lightning接口数据线的走线孔（图40）。请注意如果不是在将Workplane设在底边斜面的状态下制作Box的话，有可能会掌握不好角度。而将标尺设在底边斜面的角上则可以直接输入相对于标尺原点的位置数值，这样就可以精准地确定走线位置。

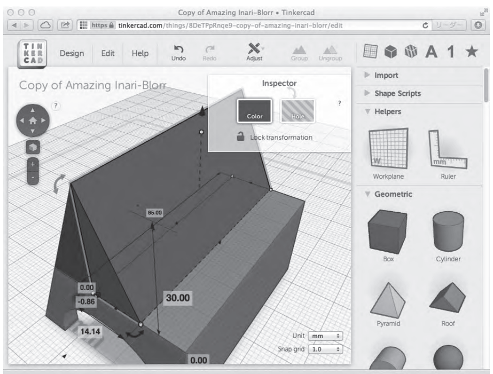
▼ 图33　复制支撑部分



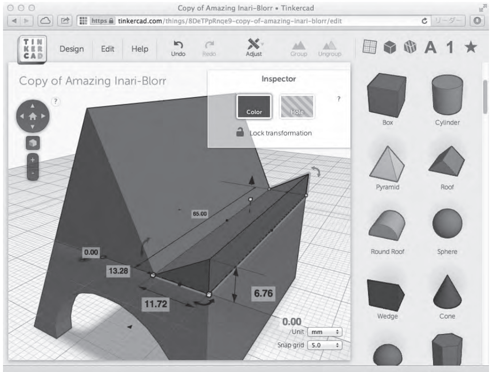
▼ 图34　制作iPhone的靠背斜面



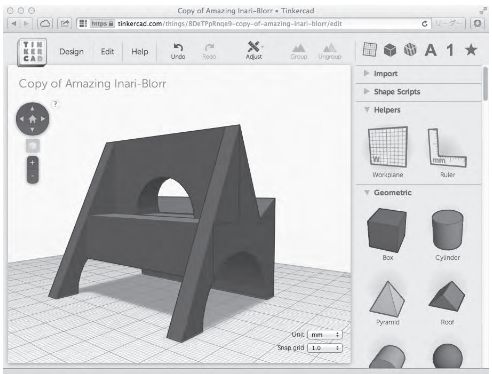
▼ 图35　扩展至幅宽



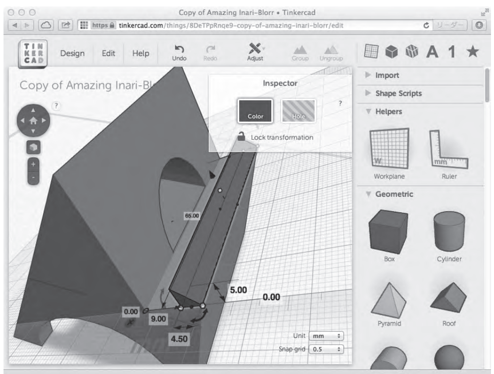
▼ 图36　制作支撑底边的部分



▼ 图37　开孔并合并



▼ 图38　制作挡板



最后，靠背部分最上面的尖顶可以不要，所以从最顶端切掉10 mm左右（图41）。将Workplane设置在与任何面都不重合的位置上，这要通过将机座垂直向下移动50 mm（拖动Z轴的黑色三角锥的手柄）来实现，然后使用足够大的Box开孔来实现切削。

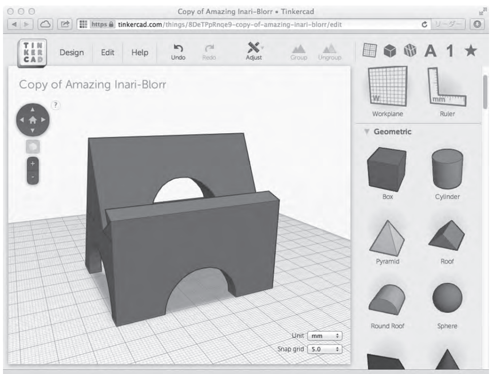
完成！

笔者想在模型上先验证一下是不是真的能承载iPhone5，所以使用Design→Close从机座的设计转到操作面板，另外新建iPhone5的设计模型。这个模型省略了圆角和按键的设计，只是做了一个简单的板状，但严格控制了外形尺寸（图42）。创建完成后使用Edit→Copy复制该模型，然后再次回到iPhone机座模型处。

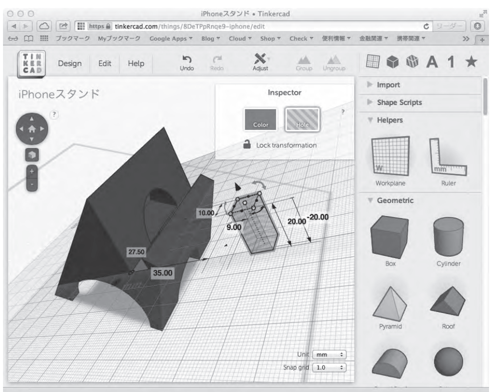
在靠背斜面上设置Workplane，粘贴拷贝的iPhone，测试其与机座的组合效果。当然，这里如果也使用标尺的话就可以准确地确定位置（参照图21、图22）。虽然有点像牌位（笑），但基本上iPhone机座的3D建模算是完成了！

笔者已经迫不及待想打印出来了，但这是还需从Design→Download for 3D Printing所弹

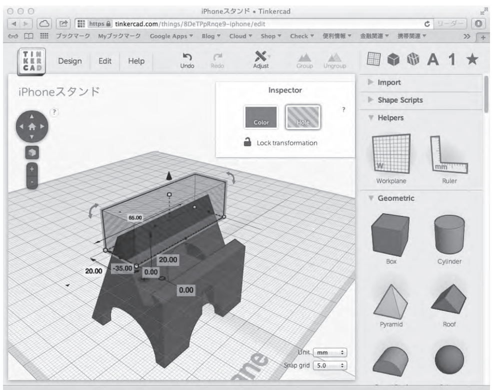
▼ 图39　切掉前面部分



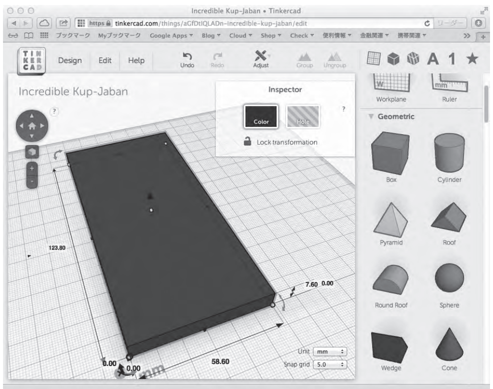
▼ 图40　开出Lightning接口数据线的走线孔



▼ 图41　削去尖顶



▼ 图42　制作iPhone的形象



出的对话框中选择“.STL”，将模型下载到本地（图43）。实际的3D打印将在下一章中具体介绍。

改进和修复Bug

实际上我们在这里介绍的iPhone机座的模型还存在缺陷和需要改进的地方。打印出来以后就会发现有一些不如人意的地方。如果用编程来做比喻的话，这时才刚刚完成了编译和创建，功能上的Bug要在实际打印出来后再修改。

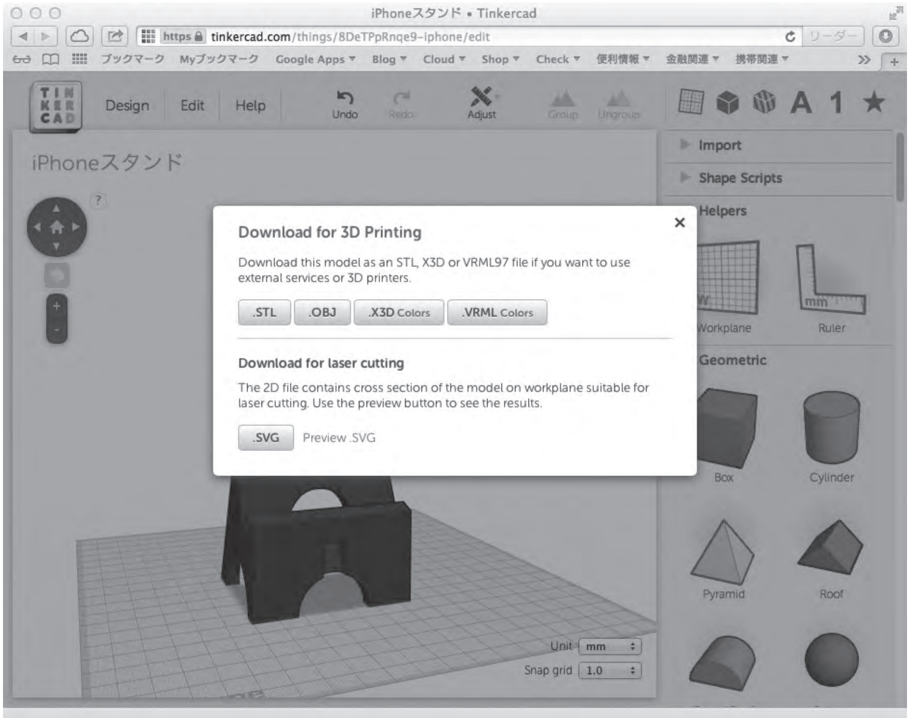
制造业所谓的Atom领域，正是通过建模修改与实物验证，也就是反复地改进模型和给实物打补丁才逐渐接近完成的。以往这样的过程在经济上花费很高，但现在随着以3D打印机为代表的、数字工程机械及快速成型设备的产生和完善（虽然不够完美），这些专业的建模过程对于我们已经不是那么遥不可及的了。

一定要在3D-CAD上练习

以往的3D-CAD是专业人士的工具，功能强大，需要相当丰富的经验才能完成建模。而且价格也在数百万日元~1千万日元（约合60万人民币）左右。但是随着Replicator及RepRap、Ultimaker等3D打印机，以及新手也能很容易入门的免费3D-CAD的出现，3D打印已经越来越简易，普通用户也只要稍加练习就能构建出不错的模型。

所以请您务必要尝试一下这些3D-CAD，它们一定会给您带来全新的体验。

▼ 图43　下载为本地数据



尝试用3D打印机来打输出

第4章

文/山田齐　工房Emerget　Http://www.emergeplus.jp Twitter @emergeplus
译/陶旭

第三章中我们以iPhone机座为例介绍了3D模型。本章，笔者将使用积层方式3D打印机来打印iPhone机座，并解说打印的流程和要点。笔者使用的3D打印机是Ultimaker①（如P.18的照片17）。

积层方式3D打印的流程

如果3D打印能像2D打印那样，把完成的模型直接从PC机上的3D-CAD输出到3D打印机的话，那就简单多了，但现实情况要复杂一些。我们首先需要用3D打印机专用软件读取STL文件，然后通过各种加工和设置，将模型转换成积层模型，并生成3D控制码，在这之后才能输出到3D打印机。

这种专用软件就是所谓的切片软件。Ultimaker的标准切片软件是Cura②（图1），那么我们就以它为例来讲解切片软件。

积层方式3D打印的流程如下。

① 将3D模型另存为3D打印机所支持的文件格式。
② 使用切片软件将模型转换为积层模型。
③ 通过切片软件生成成3D控制码（G代码）。
④ 使用3D打印机打印。

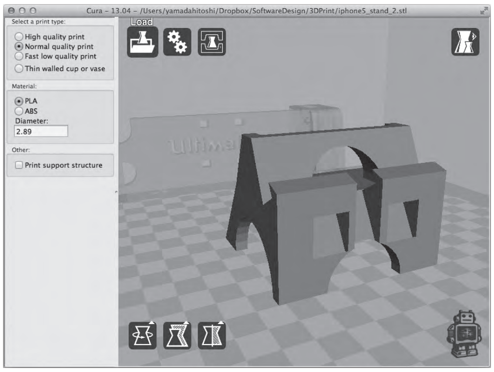
① http://www.ultimaker.com/
② http://daid.github.io/Cura/

① 将3D模型另存为3D打印机所支持的文件格式

开源环境中比较受欢迎的廉价3D打印机，基本都是由RepRap3D打印机派生出来的。而RepRap系列所使用的文件格式都是STL（Standard Triangulated Language）格式，因此我们使用的3D模型软件也必须支持STL输出。笔者在第三章里介绍的3D-CAD中，除了SketchUp Make以外都默认支持STL输出。而SketchUp Make，也只需要从扩展仓库（Extension Warehouse）③上下载

③ http://extensions.sketchup.com/en/content/sketchup-stl

▼ 图1　切片软件Cura



STL输出插件[④]即可。本章我们使用的Ultimaker就是基于STL文件设计的，因此不存在格式问题，直接将第三章中由TinkerCAD输出的STL文件保存在电脑中即可。

② 将模型转换为积层模型

积层方式打印的特点是，像医院的CT扫描那样，通过断层扫描获得三维数据，并逐层用树脂构筑起来。

首先我们需要设定3D打印机的工作条件（例如内部充填树脂的密度以及单层厚度等），一旦找到了合适的条件能够确保打印稳定，之后就很少会再做修改。然后，切片软件会导入这些条件来生成积层数据。

③ 转换成3D控制码（G代码）

下面我们需要生成G代码（图2）。多数切片软件会在转换积层模型的同时自动生成G代码，所以不需要特别操作，Cura也是如此。G代码是数控（NC，Numerical Control）机床领域中标准的控制码。

④ 使用3D打印机打印

接下来就只剩最后一步，把G代码传送到3D打印机进行打印。补充一下，Cura还有暂停功能，因此可以更换不同颜色的树脂纤维来完成双色成型。

成品越大打印耗费的时间越多，需要几个小时的情况比比皆是，我们也只能忐忑不安地等待打印的完成。

大多数3D打印机都是通过USB（或是WiFi）连接从PC传送G代码来实现3D打印的。在这当中也有可以将G代码文件写入SD卡，再将SD卡插入打印机中独立打印的机型。

Ultimaker也有独立系统控制器的选项，笔者通常使用这个功能来完成3D打印。它的优点在于，同一成品可以连续打印，而不需要经过PC机和切片软件的重复处理。而且由PC机控制的打印，有可能会因电脑进入休眠而中断输出，有时操作系统和切片软件突然死机还会造成打印机无法控制的情况，而独立系统控制器则可以随心所欲地暂停和重启打印。笔者使用这一功能的另一个原因是，放PC机的桌子和放3D打印机的桌子离得较远，USB线不够长……（苦笑）。

免费切片软件

在掌握了这一连串的操作流程之后，我们来看看切片软件。切片软件是积层方式打印机的关键所在，其与3D打印机的配合程度是影响

▼ 图2　G代码示例

iphone5_stand.gcode

```
;TYPE:CUSTOM
M92 E865.888000
M109 T0 S220.000000
T0
;Sliced /Users/yamadahitoshi/Dropbox/Software
Jun 2013 14:06:54
;Basic settings: Layer height: 0.2 Walls: 0.8
;Print time:     1:01
;Filament used:     4.07m    34.72g
;Filament cost:  Unknown
G21         ;metric values
G90         ;absolute positioning
M107        ;start with the fan off
G28 X0 Y0   ;move X/Y to min endstops
G28 Z0      ;move Z to min endstops
G1 Z15.0 F180 ;move the platform down 15mm
G92 E0                    ;zero the extruded le
G1 F200 E3                ;extrude 3mm of feed
G92 E0                    ;zero the extruded le
G1 F9000
M117 Printing...
;LAYER:0
;TYPE:SKIRT
G1 X66.8 Y83.668 Z0.3 F9000.0
G1 F2400.0
G1 E4.5
G1 F9000.0
G1 X66.8 Y113.668 Z0.3 F1200.0 E5.0488
```

④ Cura切片机软件可以在菜单中选择文件→导出→3D模型，将模型保存成.DAE格式。这样即便没有插件也可以读取模型数据。

成品质量的重要因素。现在免费的切片软件有很多，笔者只能挑选其中几款来介绍。除了笔者介绍的以外还有很多，用户可以根据操作的难易程度和对成品品质的要求来选择相应的切片软件。

（1）Skeinforge+ReplicatorG ⑤

这款是廉价开源硬件3D打印机的先锋——CupCakeCNC的切片软件。使用Skeinforge生成G代码后再通过ReplicatorG将G代码传送到3D打印机进行打印。虽然它是早期面世的切片机软件，但能够进行精细的设计也让它得到了许多用户的青睐。

（2）Kisslicer ⑥

这是RepRap很受欢迎的切片软件之一。这款软件的初始界面上有很多设置项目，让人眼花缭乱，用户可能需要很长一段时间才能习惯。

（3）Cura

这是本章所使用的3D打印机Ultimaker的标准切片软件。笔者觉得这款软件的UI比较直观易懂。在quickprint模式下只需点击3次即可开始打印。

Cura功能概要

在进行3D打印之前，笔者先介绍一下本次使用的切片软件Cura的功能。这里不做具体赘述，仅简单介绍其最基本的内容。

⑤ http://replicat.org/download
⑥ http://kisslicer.com/

Cura有quickprint和full settings两种模式，分别可以在Tools菜单里切换。在预览画面中有点击3次即打印和扩大缩小、转动后打印等图标，其UI非常简单灵活。

quickprint模式

quickprint模式正如其名，这种模式下基本都是由Cura来处理3D打印。用户只需选择打印质量、使用的是ABS材料还是PLA材料并指定是否需要支撑材料即可，非常简便（图3），笔者的很多打印都是这样直接完成的。

full settings模式

此模式下可以进行非常详细的设置，如打印喷嘴的移动速度，打印物内部的树脂充填密度和打印质量等。这里不再赘述，仅展示出Basic选项卡的截屏画面（图4）。

预览画面

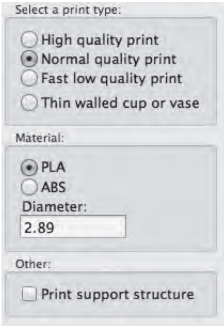
在预览画面的左上方排列有通过顺序点击即可打印的3个图标（图5），即Load（文件路径）→Prepare（转换沉积数据）→Print（打印）。进行Prepare操作时，画面下方会出现转换进度条、预计打印时间、材料使用量等信息（图6）。另外在打印过程中，也会实时显示温度及打印进度等信息，用户还可以暂停打印后挪动打印台和挤压机（图7）。

预览画面上还具有模型的扩大/缩小（Scale）、转动（Rotate）、镜面（Mirror）的功能，可以将原模型扩大（或缩小），或者为了避免悬空而将模型横放等，这些功能都设置在画面左下方（图8）。图9为转动模型时的状态。

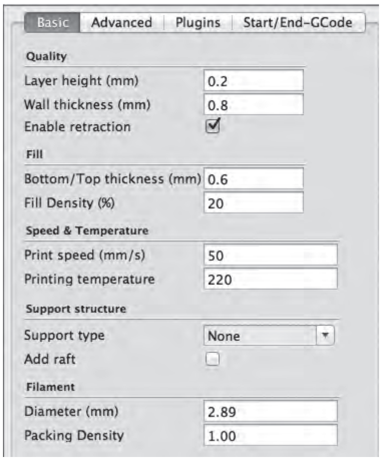
预览画面上还有一个非常厉害的功能，那

就是右上方的viewmode图标（图10）。Normal显示模型的普通状态；Transparent显示模型透明状态；X-Ray则会着色标记出3D模型上有问题（没有校准）的地方；Overhang会将悬空部分着色显示以便用户确认是否需要设置支撑材料（图11）；Layers则可查看沉积模型任意高度的断面。在打印模型前，这些功能可以帮助我们检查出模型的缺陷所在，十分好用。

▼ 图3　quickprint 模式的菜单



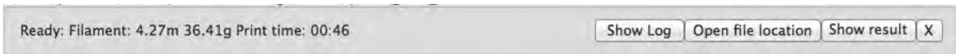
▼ 图4　full settings模式的菜单（Basic选项卡）



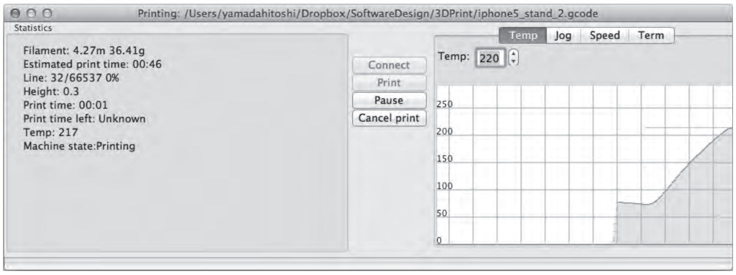
▼ 图5　Load、Prepare、Print各图标



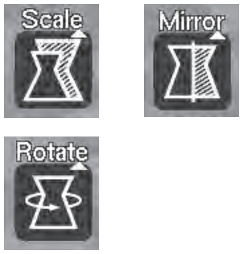
▼ 图6　Prepare状态栏



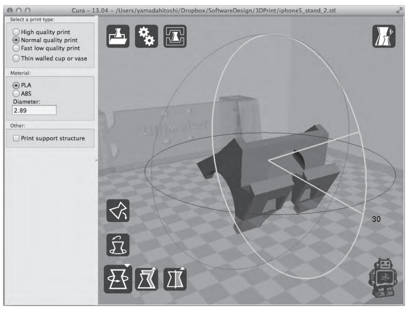
▼ 图7　Print对话框



▼ 图8　Scale、Rotate、Mirror各图标



▼ 图9　将模型Rotate的状态



▼ 图10　viewmode各图标



3D打印iPhone机座

前文叙述了那么多，下面我们就开始使用Ultimaker和Cura来打印iPhone机座吧。首先将TinkerCAD制作好的模型以STL格式下载到PC机，然后单击Load图标，选择下载好的iPhone机座的STL文件（图12）。

完成加载后，设置打印条件。由于Cura和Ultimaker能很好地兼容，所以我们使用quickprint模式打印，只需选择打印质量和ABS（PLA）材料以及确认是否需要支撑材料即可。本次打印选择了“Normal质量”“PLA”“没有支撑材料”。完成设置后点击Prepare图标将积层模型转换为G代码。此时，在画面窗口下方会出现状态栏，并开始转换。转换结束后在预览画面上会显示积层数据（图13）。

拖动右侧的滑动条可以显示内部的层面，这个功能可以用来检查内部的充填密度及外壁的厚度等是否存在问题（图14、图15）。

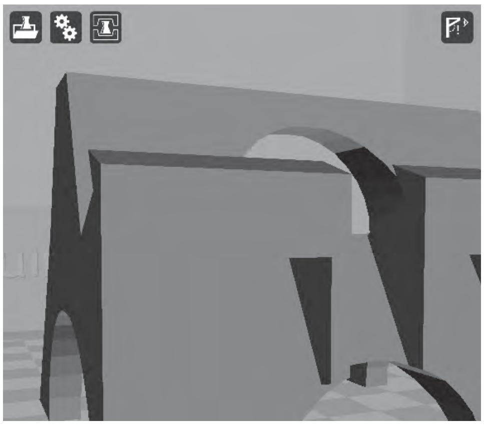
没有问题的话就可以开始3D打印了。在通过USB连接打印时，点击Print图标并在弹出的对话框中点击Print按键。

不知是Cura的固有设计还是Bug，通常即使我们点击了打印按键，加热器的温度也总是维持在0℃不变。这时我们需要先在Temp选项卡中输入PLA的推荐加热温度220℃，再点击Print键，这样加热器才开始升温，达到220℃后就会自动开始打印（照片1）。

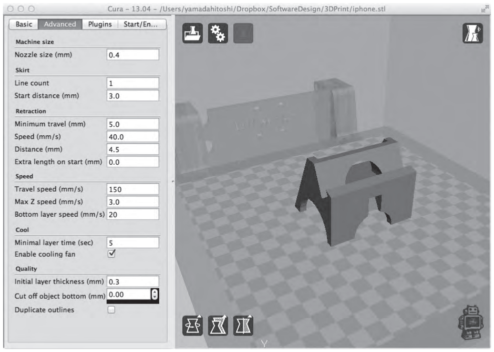
本次我们以Cura默认的速度50mm/s来打印，大概需要2个小时（照片2）。

用PLA材料制作的成品会牢牢地粘在打印台上，很难取下，贴在打印台上用于防护的胶带也很容易弄破。而且，如果打印的作品比较薄，还有可能划伤手，因此用户需要特别小心。

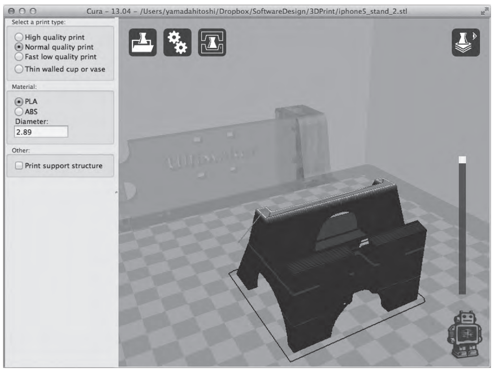
▼ 图11　overhang显示（对象部分以红色显示）



▼ 图12　加载了iPhone机座



▼ 图13　显示积层数据



相反，ABS材料则因为太容易脱离打印台而导致在打印过程中发生位移，经常打印出废品来。

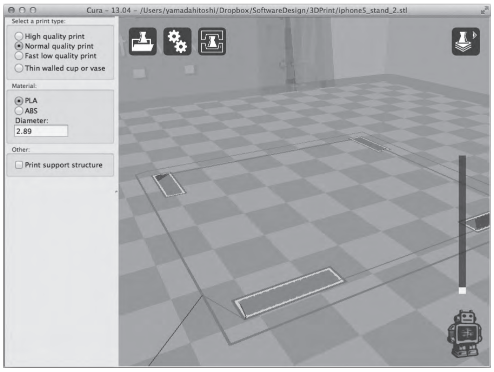
iPhone机座打印成功后，边边角角上还带有很多纤维屑（如P.18的照片18所示）。目前的积层方式打印总会出现这样的情况，用户还需要使用钳子、刀具、锉刀等工具进行精整。

下面我们来检查悬空部分。在跨度比较长且没有支撑的部位还是发生了一些下垂（P.18的照片19），但这对于功能没有什么影响，所以我们不必对其特别处理。

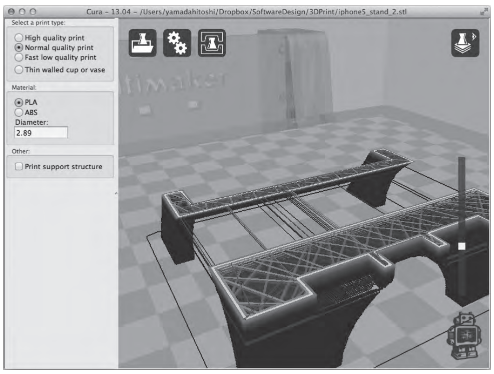
这样iPhone5的机座就完成了（照片3）。现在我们把iPhone5放上去试试，横着放的时候完全没有问题，可纵向放的时候……虽然可以放上去但重心稍向后倾，轻轻一推就往后倒了（照片4），所以需要把撑脚再加长一些。

另外虽然设计了Lightning的走线通道，但每次必须拔掉插头才可以放上去，非常不方便，

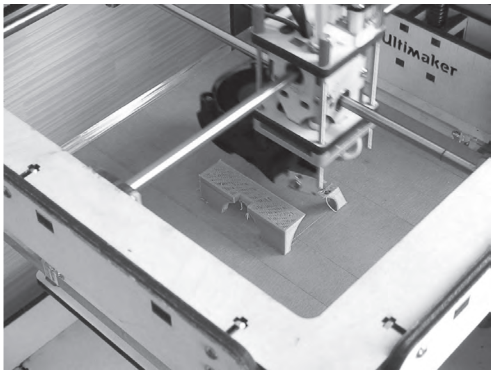
▼ 图14　截面的最下层



▼ 图15　内部截面



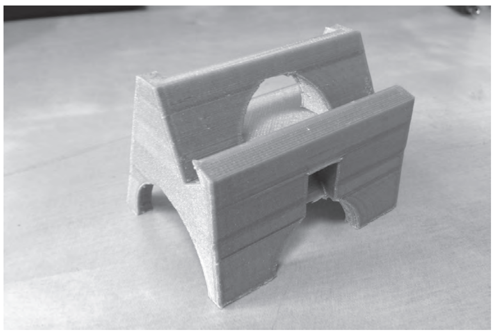
▼ 照片1　打印中的iPhone机座



▼ 照片2　打印结束的iPhone机座



▼ 照片3　完成的iPhone机座（试制品）



这点也需要再改进。而且纵向放置时机座还会遮挡住扩音器，听音乐的时候声音有些闷，所以最好在正对扩音器的地方开个孔。

快速成型的魅力

现在，经过试制我们很快就找到了iPhone机座的问题。使用TinkerCAD修改模型并重新打印，这个过程笔者不再赘述。最终成品的效果见P.18的照片20、照片21以及本页的照片5、照片6。

▼ 照片4　将iPhone5纵向放置（试制）



▼ 照片6　将iPhone5纵向放置（完成）



首先将撑脚加长了6 mm以防止后翻。把Lightning接口线通道做成沟状以便于插着数据线就可以直接纵向放置。同时还在扩音器处开了孔，改善了声音发闷的问题。另外，在iPhone左侧麦克风处也开了孔，以便于进音。

第二次的打印非常完美，笔者终于完成了想要的机座。另外，笔者已经将这个3D模型利用TinkerCAD共享在Creative Commons的BY-SA上[7]了，欢迎大家直接使用。

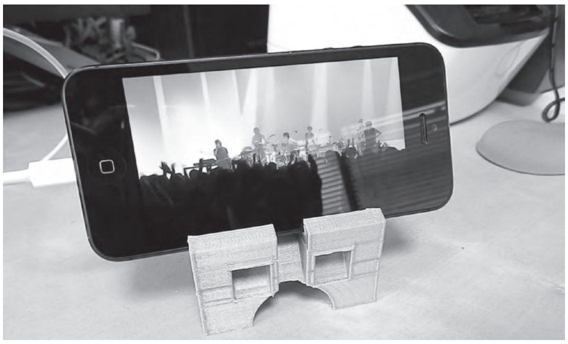
以3D打印为代表的快速成型的魅力在于即便失败了也可以立即重做，而且不用耗费太多的成本。其实，3D打印一般很少一次成功，这些失败的经历可以让我们掌握机床的特点并积累设计经验，有很多益处。并且我们在一次次反复制作的过程中还可以体验到自己动手制作产品的乐趣。

3D打印机的相关维护

最后我们再来谈谈对于3D打印机非常重要的维护问题。

⑦ http://tinkercad.com/things/hOY27ARjFzb-iphone5-stand2

▼ 照片5　将iPhone5横向放置（完成）



3D打印机也属于机床的一种，为了维持其正常运转需要进行日常维护。特别是廉价的积层方式3D打印机，它面世的时间不长，而且是靠很多零件组装起来的，稍有不妥就会出问题，因此在这里列举一些维护要点。

X、Y、Z各轴

为了使3D模型能够按照设计去打印，必须保证决定位置的X、Y、Z轴能流畅运转。Ultimaker的X、Y轴是皮带轮轴，Z轴是螺杆轴，并由步进电机驱动。

当模型发生歪斜时，需要检查皮带张力是否适当，轴润滑剂是否正常，是否由于筐体不正而造成轴歪斜等。

打印台的水平度

打印时打印台必须保持水平才能正确打印出成品，如果打印台倾斜则会导致纤维不能正常固定、挤压喷嘴碰触打印台、紊乱（步进电机失控空转）等问题。

机器上应该都有调节打印台水平度的配件，如调节螺栓等，在打印出现问题时可以进行相应的调整。

挤压机

挤压机会用加热器融化树脂纤维，并将其从喷嘴处呈细长丝状挤出，因此里面装有控制温度的温度传感器，但如果传感器出现问题则会导致温度错误而不能挤出纤维。

如果纤维不能正常融化，则需要检查传感器是否损坏或脱落。

传送纤维的装置

虽然传送纤维的装置安装的位置因打印机机型而异，但通常是由夹住并推送纤维的滚轮部分和通过齿轮驱动这个滚轮的步进电机组成的，如果这个结构不能流畅运行就会造成纤维的传送量与计算值不符。而且需要注意夹住纤维的力度不能过松也不能过紧。如果纤维不能正常挤压则需要检查这个装置。

另外，有时候纤维也会自己缠在转轮上转不起来，这时也要检查一下这个装置。

他们看3D

“3D打印将无处不在，并让每个人都能成为创客。”

——3D Systems公司总裁Avi Reichental

“为什么我离开《连线》，因为3D打印是一件比互联网更大的事情！”

——《连线》杂志前主编克里斯·安德森

“3D打印从表面上看是颠覆了产品的制造方式，但从本质上看却是传统产业面对的挑战和机遇。从某种意义上说，3D打印对现有企业而言，要么是天使，要么是魔鬼。”

——海尔集团董事局主席、首席执行官张瑞敏

“万科三年后将3D打印出房子。”

——万科集团董事会主席王石于2014年“万科公开讲坛”

“在我国，3D打印之所以广受关注，是因为它与创新紧密相连。”

——中国工程院院士卢秉恒

第5章 如何选择3D打印机

文/坪井义浩 ytsuboi@gmail.com Twitter @ytsuboi
工房Emerge+ 山田齐 http://www.emergeplus.jp Twitter:@emergeplus
译/孟海川

到此为止，关于3D打印机已经介绍了很多了，但是，该如何选购一台合适的3D打印机呢？想必还有很多人会感到困惑吧。本章将以购买率较高的3D打印机为例，具体介绍一下3D打印机的选购。

Replicator 2

消费级3D打印机中，Makerbot公司的产品是一个很好的选择。如第1章中所述，Makerbot公司的基础业务是RapRap项目，该公司从2009年开发销售Cupcake CNC这款机器以来，一直被认为是个人3D打印机的代表生产商。继Cupcake CNC后，Thing-O-Matic型号的机器也开始发售，但这两种机器是搭配组装的，因此有很多用户觉得组装和调整机器性能方面很费劲，与期望有所差异，更有的人觉得根本不好用。因此，在3D打印机的使用体验上，制造商还需要进一步努力。

鉴于这种情况，Makerbot公司推出了Replicator，作为无需组装的成品机型开始发售。进而，2012年发布的Replicator 2（P.18照片22），身为成品机的同时由开源转为非开源，这一转变也广受争论。

现在Makerbot公司的主力产品为Replicator 2和最近销售的Replicator 2X（照片1），开源与否的问题姑且不说，不论如何，Makerbot的打印机都是非常热销的。

Replicator 2的魅力

Replicator 2的魅力在于，打开包装箱取出打印机就可以立刻使用了。Replicator 2还搭载了可以独立运行的控制标准。

从箱子里取出打印机，连接线路、电源后启动机器，初始设定的程序也随即启动。通过这个程序，可以调整打印台的位置（高低等），以及把纤维插入到挤压机中。如果不太清楚这个过程的话，可以到YouTube上观看安装视频[①]。还在犹豫是否购买的顾客也可以去看看这个视频，事先了解下这到底是怎样的机器。

① 从箱子中取出：http://www.youtube.com/watch? v=ENtFOM-2160
初使设置：http://www.youtube.com/watch? v=KYGjICo-fV4

▼ 照片1 Replicator 2X



附带的SD卡中有多种3D打印数据。组装完成后，从这些数据中选择想输出的数据。

易发故障

虽然Replicator 2使用方便，但是也会发生一些小故障。如果刚使用时发现有错误，请联系售后服务。若是加热器不良导致的故障，可以进行零部件调换。另外，笔者身边也有因喷嘴处树脂泄漏而忙于和客服人员进行交涉的例子。

另外，在使用Replicator 2过程中，若挤压机（Extruder）出现松动，Thingiverse上有用户上传的改良部件数据[②]，可以自行下载打印。Makerbot也以8美元的价格销售替换部件的数据[③]。不管哪一种方法，都需要进行3D打印。除此之外，常见的问题还有打印台不平整等。Replicator 2仅可使用PLA树脂纤维，挤压机只有一个。

新发售的Replicator 2X，增加了支持的打印材料，在PLA的基础上增加了ABS树脂，并且配备了遮光罩，所以使用ABS树脂打印也可以顺畅地进行。挤压机增加为2个，并且改善了Replicator 2的挤压机和打印台不平整的问题。这些改进，使得Replicator 2X更加有吸引力。

虽然Replicator 2经过多番改良，但还没有达到无故障运行的程度，多少会存在些问题。因此希望读者能认识到，想要使用这款产品就要不断试错才能顺利打印。另外，客服方面，Makerbot公司的客服使用的是英语服务。

用户团体

有一个名为Japan Makerbot User Group[④]的用户团体，该用户团体是由很多博客作者创建的，里面有各种有用的信息，笔者经常会参考这些信息。

② http://www.thingiverse.com/thing:42250
③ http://store.makerbot.com/drive-block-hardware-kit.html.
④ https://groups.google.com/forum/?hl=ja&fromgroups=#!forum/jmb-ug

关于Replicator的价格，Makerbot公司官网[⑤]上Replicator 2销售价格为2199美元，Replicator 2X为2799美元。

Ultimaker

Ultimaker（P.18照片23）是荷兰Utrecht Fablab开发的RepRap系列中的开源3D打印机。当时他们对于市面上各种3D打印机的速度或品质多少觉得不理想，于是决定采用挤压机和纤维传送装置分离的形式，在打印机的功能开发上颇下工夫。

Ultimaker的机体框架采用了激光切割的胶合板，非常牢固。标准打印原料是PLA，同时也支

⑤ http://www.makerbot.vom

▼ 照片2　Ultimaker Robot



持ABS。但纤维的直径是3mm，并非当今的主流。

笔者已经购入Ultimaker，去年在纽约参加Maker Faire 2012会议时，曾得到一个Ultimaker Robot的样品（照片2），后来又参考了*Make*杂志发行的一本*Make* : *Ultimate Guide to 3D Printing*的专刊[6]，最终决定购买Ultimaker。Ultimaker是一款性价比非常高的3D打印机。打印速度、质量、加工尺寸等都很令人满意。如果自己组装的话，以1194欧元（约人民币1万元）的低价格就能买到（不包含自选配件、运费、关税等）。笔者也是从Ultimaker网站[7]购入元件，然后自己组装的（通过Maker Shed[8]也能购买）。

是否能完成的组装

RepRap系列的3D打印机中，很多都需要自行组装。在线指导手册大部分是英语版，想必不能顺利组装的用户也大有人在。Ultimaker送过来的配套元件是改良版，与在线指导手册中纤维传送装置的构造完全不同，也没有电气布线的指导图片连接。所以，组装Ultimaker，对于不是非常擅长机器组装的人来说，是很困难的。若用户不想自行组装，可以购买1699欧元（约人民币1万4千元）的Ultimaker，这其中会包含组装服务，还会赠送独立控制的组件。对于不擅长机器组装用户，这个方案还是比较划算的。

笔者的Ultimaker使用近半年了，基本上没有太大的故障，顶多是需要偶尔调平打印台。网上流传的挤压机喷嘴问题也没有发生过。总体来说，这款机器还是很棒的。

Ultimaker采用的是Cura标准切片软件，组装完成后直接运行测试软件，不用做任何调整，仅需点击3下，就能打印Ultimaker Robot的样机模型了。这一点也是我选择Ultimaker的理由之一。

RepRap atom

本期特辑曾多次提及RepRap（照片3），介绍消费级3D打印机，基本上也是绕不开RepRap的。RepRap的特征之一是“自我复制”机能。RepRap是开源打印机，GPL许可里可以查到其设计信息。因此，用RepRap可以打印出RepRap的树脂零部件，组装后就能制造出一台新的RepRap机器。

如果身边没有购入过RepRap或3D打印机的朋友，也可以从RepRap Pro[9]等网站购买配套元件并自行组装。

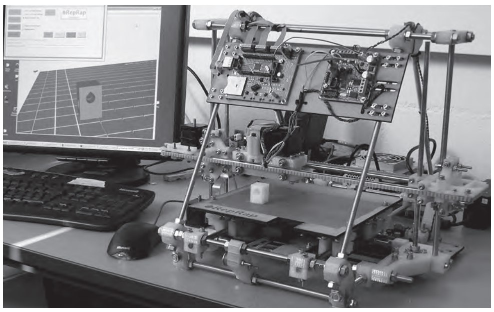
RepRap机型繁多，到底该选哪一款比较好呢？针对日本用户，笔者推荐RepRap atom这一款（P.18 照片24，由@arms22提供）。

自行组装必备信息

RepRap atom是日本RepRap团体（RepRap

⑨ http://repprapro.com/

▼ 照片3　RepRap Version Ⅱ “Mendel” machine



⑥ http://makezing.com/volume/make-ultimate-guide-to-3d-pinting/
⑦ http://shop.ultimake.com/
⑧ http://www.makershed.com/Uloimaker_3D_Printer_P/mkum1.htm

Community Japan[10]）发布的3D打印机。不仅限于atom，大部分的RepRap都需要自己组装。若不想购买指定的配套元件，就需要自己收集工具然后组装，这种方式非常有难度。配件只能通过eBay购买，而且很容易断货，所以不得不一直留意网上的动态。

以挤压机为例，就需要组装如照片（P.18照片25，由@arms22提供）中的树脂零件、脉冲电动机（stepping motor）、螺丝、螺母、垫片、弹簧等。以上信息是由RepRap Community Japan提供的。如果不参考这个团体的信息自己去做的话，是很难实现的。

打印的质量

读到这里，一定有读者认为开源打印机，输出质量也不会有多好。可是，别忘了Makerbot也是从RepRap里演变出来的。RepRap的打印效果不仅不错，而且各方面的性能都有保证。另外，因为是开源打印机，所以可选择的控制软件也很多。就像刚才介绍Ultimaker时提到的Cura就可以和RepRap一起使用。

截至本稿执笔，最新的atom[11]是RC版，可以自行组装，也可以花费147700日元（约人民币8500元）购买包含组装指导服务（组装workshop）的成品套件[12]，这样的话难度会大大降低。如果不需要组装指导服务，仅购买配套元件[13]的话是129800日元（约人民币7500元）。不过，组装指导服务仅贵了10500日元（约人民币1000元），如果想购买RepRap的话，还是比较推荐包含组装指导服务的方案。

⑩ https://sites.google.com/site/reprapcommunityjapan/
⑪ https://github.com/hironaokato/atom_rc
⑫ http://my.ebshop.info/
⑬ http://genkei.thebase.in/

小结

现在，消费级3D打印机的机能大致都差不多。最近大家甚是期待光固化成型技术的打印机器，但是离这种机器的普及还需要一段时间。就现状来说，如果购买3D打印机的话，发生故障时有能够探讨故障的其他用户或团体是关键所在。

从这个层面来说，日本使用Replicator 2的人居多，便于搜集相关信息。RepRap atom虽然刚发布了Beta版和RC版，用户还是不很多，但是很多人3D打印之旅的初始选择都会是atom。就笔者所知，Ultimaker是3款当中用户最少的一款。但是就笔者周围使用3D打印机的朋友来看，这3款3D打印机当中Ultimaker是故障率最少的一款，个人对它的评价还是很高的。

除去详细介绍的这几款打印机之外，还有其他低价3D打印机机型也不错，例如3588美元（约人民币21700元）的3D系统的CubeX[14]。最近，价格更为低廉的Cube也通过国内代理店开始上市。

选择3D打印机的时候，大家一般都很关注商品目录的说明书和价格。但是笔者建议，要更多注意周围其他用户或者团体的体验反馈，以及国内是否有代理店售后服务等。另外3D打印机技术的发展迅猛，最好常常关注下最新的信息。

▼ 照片4　Cube



⑭ http://cubify.com/

Column 消费级全彩3D打印机

全彩3D打印，基本上都是由采用粉末固定积层法的ZPrinter在唱独角戏，消费级3D打印机领域中几乎没有。但从今年开始，这些情况正在悄然发生变化。

例如，botObject公司[15]开发的ProDesk3D打印机，使用了5色PLA热熔解积层法，可以实现全彩打印，甚至可以用PVA打印出支撑材料。笔者最近也一直在关注这款机器，其价格为3429美元（约人民币20700元），确实属于消费级3D打印机的价位。

⑮ http://botobjects.com

Column 关于使用成本

如果购买3D打印机，到底需要多少成本呢？对于3D打印机来说，必要的消耗品就是纤维和打印台保护胶带。胶带不是必须的，但若使用没有加热床的3D打印机和PLA纤维的话，则可以使用胶带保护打印台。

以Ultimaker的在线商店的纤维价格为例，白色0.75kg的PLA纤维是31.5美元（约人民币200元）。3、4章中制作的iphone支架，输出的重量是30g，所以纤维的重量是0.75kg的1/25，即以165日元（约人民币10元）的原材料成本就能完成制作。

关于具体的使用成本，Makerbot公司和Replicator对大量的3D打印机运行结果报告进行了分析，根据该公司博客[16]的记载1kg的卷轴（热熔丝的一卷）能制作392个国际象棋的马。Makerbot公司使用的热熔丝为PLA，不同颜色价格也会有所不同，价位在43-48美元之间。5千日元（约人民币300元）的热熔丝能做出392个棋子，一个棋子的原料价格大概在13日元（约人民币8角）左右。

⑯ http://www.makerbot.com/blog/2012/02/24/a-matter-of-scales-how-much-com-you-print-with-a-single-1kg-spool/

行业趣闻



据《新科学家》(*New Scientist*)杂志报道，阿姆斯特丹一条运河沿岸，建筑师们正尝试只使用3D打印技术打印一座房子。如果成功，这座房子将成为3D打印出的世界最大建筑。

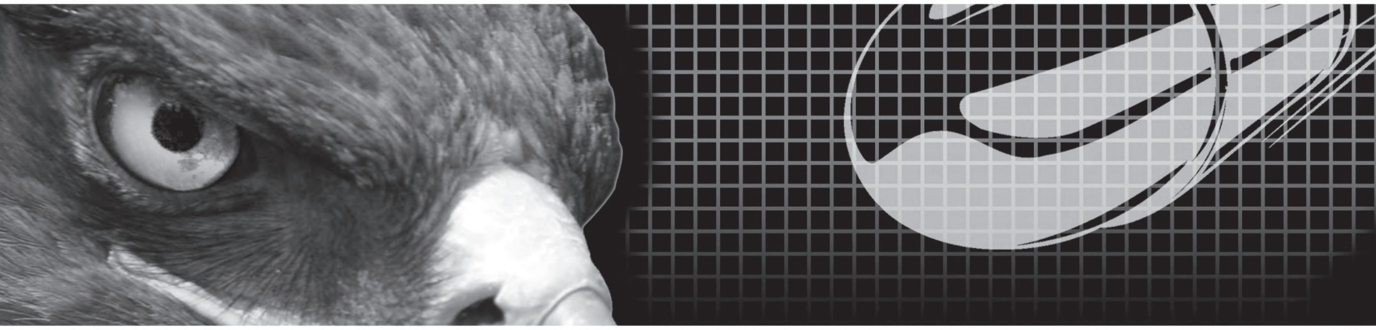
今年早些时候，当地公司DUS的建筑师们就开始着手此项工作。如果感到好奇，路过这座在建建筑时，可以花上2.5欧元看个究竟，顺便投资支持下房屋建设。

该公司使用高达6米的3D打印机KamerMaker（荷兰语“房屋建造者”），其功能与桌面3D打印机相似，可在连续层次中挤压出热塑料。

建筑师将每个房间单独打印成大模块，然后像乐高积木一样组装起来，已经完成的房间可以安装在其他房间顶部。该建筑的最终形态看起来应该会像传统的荷兰运河房屋，工期预计为3年。完成后，该建筑将作为设计博物馆向公众开放，全部12个房间将用于不同类型3D打印建筑的研究。

哈佛大学制造实验室的史蒂芬·欧文（Stephen Ervin）表示：“随着规模增加，DUS可能会面临很多问题。有些东西在桌面或实验室上运转良好，但当规模变大后，可能会突然失效。”

——摘自凤凰科技（http://tech.ifeng.com/），略有改动



以系统洞察力（眼力）驾驭软件开发

狙击Bug的技术

用社会统计、医学等其他产业的知识来看软件评测技术

银弹真的是必须的吗？

为了找到Bug，需要读源代码。顺着逻辑结构，有时使用试错法，有时借助调试功能的力量。不过，正如大家所知，现在的软件都是由庞大的文件组成的，逐个把Bug找出需要大量的劳动力，所以这变得越来越不现实了。

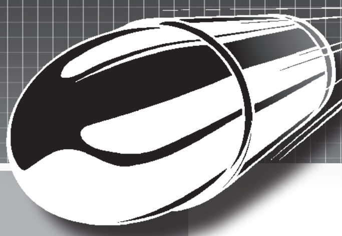
另一方面，为了应对这种状况，管理软件质量的测试和评测的技巧也在不断地进步。如果应用本特辑介绍的QI法，不用直接读代码就可以找到Bug的原因，这才是真正的狙击Bug的技术。比如把源代码用文本编辑器进行排序，ELSE IF的数量不一致、没有进行Try catch的例外处理等问题便一目了然了，马上就能找到什么地方有Bug，而且这也只不过是评测技术的一个小例子而已。如果理解了本特辑的内容，就能找到习得系统洞察力（眼力）的窍门，从而在加快软件开发速度的同时也可以提高软件质量。

●细川宣启
日本IBM股份有限公司　软件事业部
Rational事业部Worldwide Tiger组　质量工程师



狙击Bug的技术

以系统洞察力（眼力）驾驭软件开发



用社会统计、医学等其他产业的知识来看软件评测技术

文/细川宣启
日本IBM股份有限公司
软件事业部
Rational事业部
Worldwide Tiger组
质量工程师
译/芳龙

洞察软件缺陷的能力

大家好，我是本特辑的执笔者细川。笔者从事软件缺陷和Bug的研究。在业务系统开发、设计、要求分析等方面有丰富的实践经验。后来也曾在质量控制部任职。从事产品质量控制工作后，又作为质量管理技术、质量系列工具的研发者一直工作至今。

开发者的软件代码、规格说明书等在质量验证、指标测定等标准中是怎样被评测的呢？本文将对其进行详细介绍。

本文通过分析各种质量评测技术（软件评测），将阐明今后软件开发技术者应该具备的“眼力”。这里的“眼力”就是指“洞察系统的能力”。

软件成果是怎样被检查的呢？

软件测试或评测的专家如何检查开发人员的成果呢？检查的范围要涵盖成品的全部，难道他们要拿着红笔逐页查看规格说明书或代码吗？如此慢悠悠的质量负责人当今是没有的（如果有人这样做的话，反而会给开发带来相当大的麻烦）。

近来，对软件开发工程师而言，“质量中带入速度”的思考模式日益重要。他们追求在质量活动中保持省时高效的低负荷状态。因为在严峻的系统开发商业领域中，以往那种高负荷的质量评测技术，在市场竞争中已经沦为致命的缺陷。

如果能了解各种质量管理技术，知晓开发者是怎样被评价的，那么开发方就可以事前做出应对。若能很好地实现这些，甚至可以不需要质量负责人。而且，还可以获得一系列优势，诸如能够满怀信心发布产品或系统、消除系统开发后期的混乱等。这些就是作为开发工程师应该具备的“洞察系统的能力”。

眼力①社会统计学（基尼系数）可以应用于IT领域？

系统开发项目可以用社会统计学来理解和说明，你相信吗？

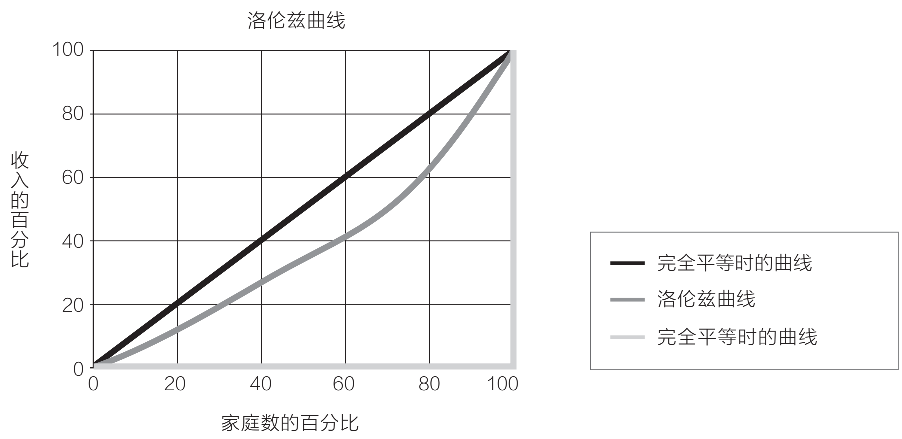
社会统计学中，用来衡量收入分配不均程度的指标叫基尼系数[1]。基尼系数于1936年由意大利统计学学者基尼设计发明，其基础是分析事物集中度的“洛伦兹曲线”基尼系数，因可以判断收入分配差异程度而名声大噪。除此以外，它也可以用于纵观某个社会中财富分配的不均匀性。

收集不受社会个体影响的统计数据，从而纵观社会整体性质，并且定期检测这些数据的变化，这就是一个比较好理解的统计实例。

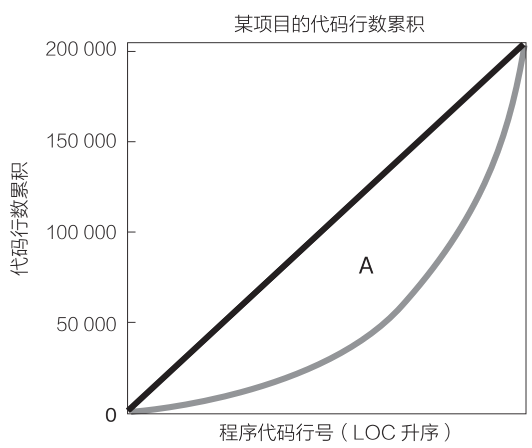
图1为表示收入分配的洛伦兹曲线例子。

IT界也能见到社会统计学中的这种形式，那就是代码行数（以下称作LOC，Lines of Code）的累积图（图2）。看吧，是不是与社会统计学中的图一样？

▼图1　表示收入分配不平等程度的洛伦兹曲线[2]



▼图2　某项目的代码行数累积曲线



系统开发项目的软件规模是可以用源代码的行数来表示的。代码的行数也可以作为估计工作量、计划立案等的衡量指标。请看图2，x轴表示程序代码行号（LOC升序），y轴表示代码行数累积，以此规则作图就可以得到图2的曲线。

（如果用社会统计学用语进行说明）基尼系数的━━线和━━线之间的闭合面积A较大，就意味着代码行数偏向于某个特定的程序，也就是说代码行数都集中在某个特定程序中了。

这和社会中收入分配不均现象的发生是一样的，也就是说有几个程序的代码行数偏大。假设所有程序的行数都相等，在行数指标上没有偏向的这种均衡状态下，━━线和━━线会重叠，基尼系数就变成了0。

在项目中预估和把握项目规模时，如果不考虑多个程序代码的行数离散和偏差等问题，只是单纯地通过计算总和及总和的平均来掌握行数，会造成与原本预估偏差较大的混乱。另外，把质量无法保证的1至2个程序从项目中选出来时，根据事先计算的基尼系数进行检查，就可以节省检查整个程序的时间，并且可以推定出有几个程序的大小不正常。

悟性较好的读者可能早已察觉到，基尼系数不一定非要用于程序的行数统计，也可以应用于注释占有率或者单纯的IF语句数等统计。

IT系统开发，可以用社会统计学中的“群体”分析技术进行分析。纵观全体，锁定并抽取出有问题的个体。这种技术甚至还可以用来比较多个项目。

社会统计学可以应用于系统开发的原因

社会统计学可以用于系统开发的评价、测定的原因是：两者在“分析人类集体行动的特征”这一角度是共通的。

换句话说，人在“开发项目”这个集体行动中，一定会如代码累积行数那样从数值上体现出的一些征兆。并且，一般情况下，个人是无法回避这些征兆的，这也是专家测定质量的最佳立足点。

无论是收入分配还是系统开发，集体在做出某项行为时，就会体现出其特有的“征兆“。能否准确掌握集体特有征兆，越来越成为开发工程师必备的“眼力”。而观察人工产品的特性、从数据中分析出其制作集体“征兆”的技术正是本文题目中提到的“眼力”。对于将成为新时代技术者的读者们而言，这将是必备武器。

眼力②使软件评测变轻松的QI法

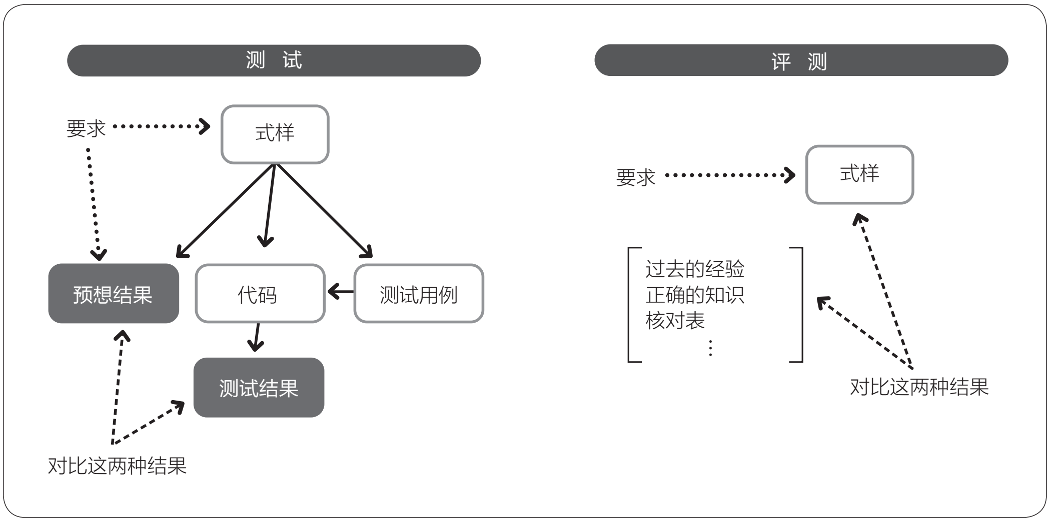
作为“洞察系统的眼力”的应用事例，这里讲一下质量评估中的“评测”。首先，讲一下为什么评测的难度较大。

评测难度大的原因

请看图3。一般的评测方法从自由度的高低、实际工数的负荷强度等角度出发，在项目中进行形式化质量评测。与对比两种结果（预想结果和测试结果）的测试不同，评测活动难以量化。基于这些原因，评测在很长一段时间内都被认为是很难适用的技术。

而且，对于大规模的项目来说，检查初期阶段的规格说明书的所有内容，逐行检查所有代码等都要花费大量的劳动力和时间，甚至会

▼图3　测试和评测的区别



成为项目成本大幅增加的诱因。

你知道QI法吗?

如何才能减小评测负荷，并把效果最大化呢?

答案是，在评测前灵活测定指标数据。

也就是说，如果能够事先掌握所有成果中的缺陷位置以及种类的话，就能锁定重点评测对象，进而提高缺陷检出效率，缩减质量评测成本。这就是所谓的QI（Quality Inspection）法。

对于进行着数十到数百个项目的企业或组织来说，经常必须对所有项目整体进行全面评测。假如可以事先锁定评测目标，就可以将有限的精力集中投入到某个目标项目中，进行有效评测。

应用QI法，不必逐页检查大量的规格说明书，而只选择检查那些看起来有缺陷和缺陷率高的地方就可以了。

检查代码时，如果知道某些代码应该有某种常见缺馅，那么就不必耗费心力，逐一检查几十个代码文件了。换言之，即在打开规格说明书之前，明确缺陷的种类和位置了。QI法的概念如图4所示。

QI法学习成本小，易掌握，便于开发方进行自检。应用此方法，质量评测负责人也可以在短时间内把握整体质量，锁定缺陷混入率高的地方，从而实现高效检查。QI法不仅局限于评测，在分析判定测试、发布等方面也是有效的。

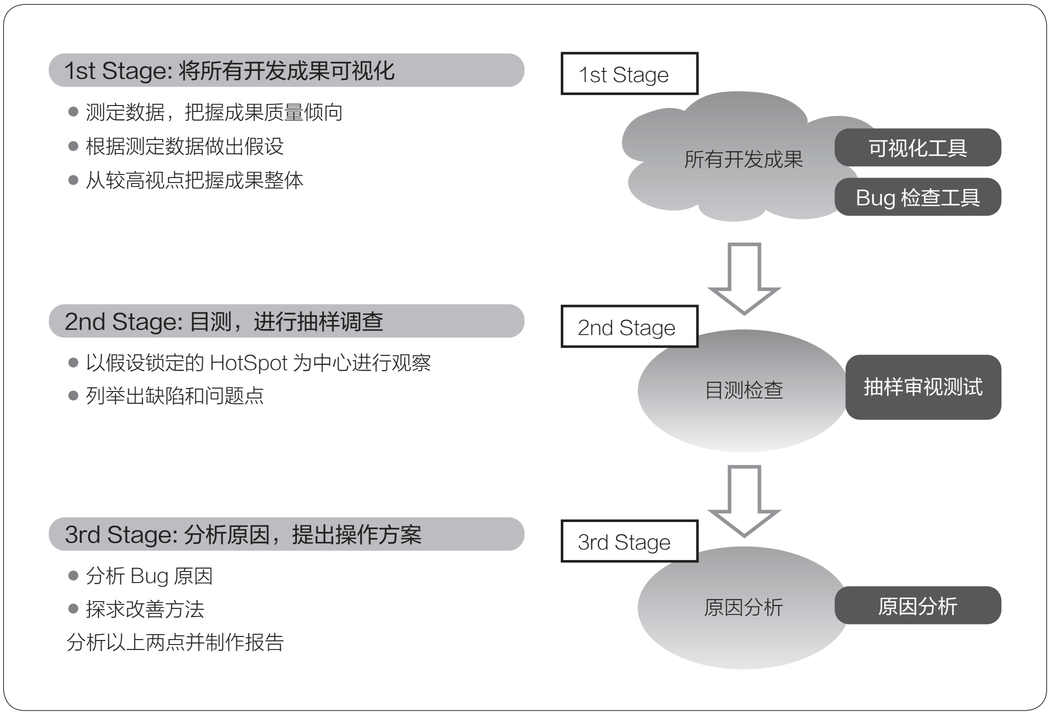
QI法是实现超高速评测的魔法?

下面就体验一下“以最快速度保证质量底线”的魔法吧。QI法擅长高效锁定Bug的位置和种类，其流程如图5所示。

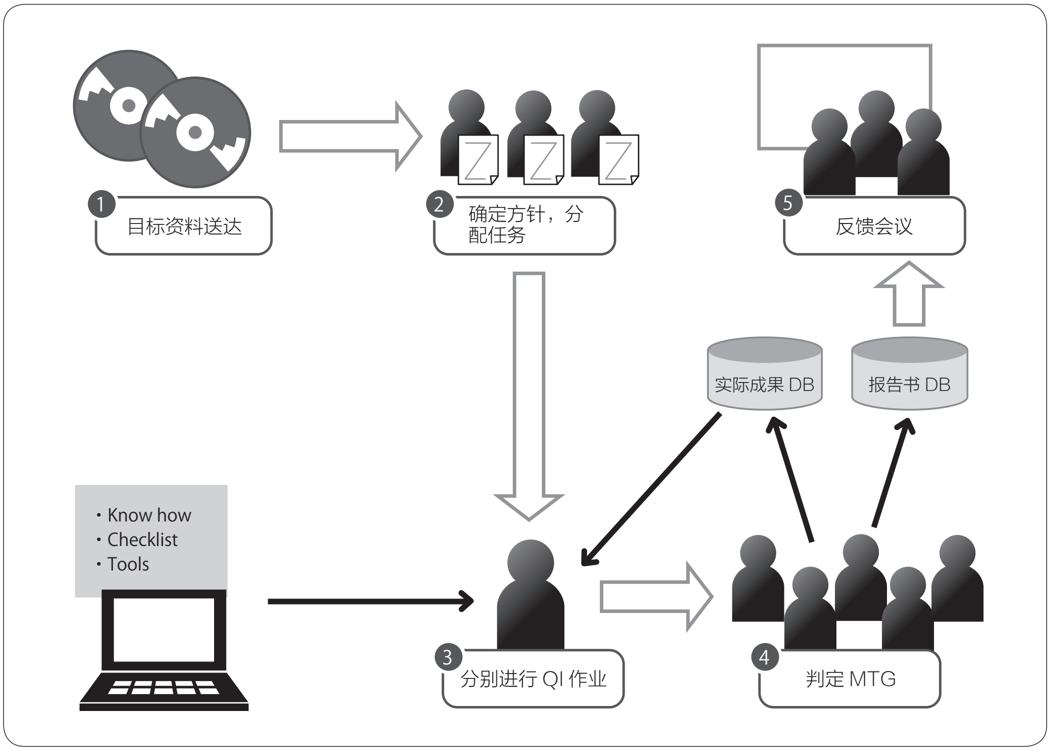
目标资料送达之前QI法已经开始了

评测负责人会领取系统开发相关资料。首

▼图4　QI法操作概要



▼图5　QI法流程



先，①接收评测对象，是整个流程的开始。特别需要指出的是，当第三方质量检测机关要求“进行质量评测，请先向检查组织提交评测对象的资料”时，QI法可以对资料送达的时间进行衡量和测定。

管理服务器上成果内容，特别是取得工作人员最新版的内容等，都会影响资料提交时间。评测专家在资料送达之前，就已经开始做出如表1所示的假设了。

资料送达时间与问题项目之间的奇妙关联

若项目符合上述假设中部分或全部内容，就意味着发生下列情况的几率很高：未进行自检、成果提出时完成度不够、开发者之间缺少协调。进而也可以判断，目标对象材料中出现表1中“预想缺陷”栏的几率也会很高。

当然，资料送达时间会因组织、项目等各个企业的基准值而不同。不过有趣的是，多数情况下，其关系会呈现为正规分布（泊松分布）(图6)。

评测对象资料的“平均送达时间”和项目管理的充分性以及团队内的沟通程度是否有关联呢？虽然研究还不充分，但与前述中收入不均衡的例子一样，若将其作为把握集体行为的测定指标，会发现它们之间似乎有着奇妙的关联。

确定方针・分配任务

接下来的②是确定方针、分配任务。这里要做出关于缺陷的假设。

资料送达后，不要从资料的第一页开始逐页检查。在开始之前，一定要对其数据进行测定。测定整体倾向的指标有流程指标、项目指标等，但首先应该用几十种测定负荷小的简单指标来掌握整体倾向。

对项目整体而言，资料检查应该如何着手？表2和图7是时间轴下的指标测定结果。

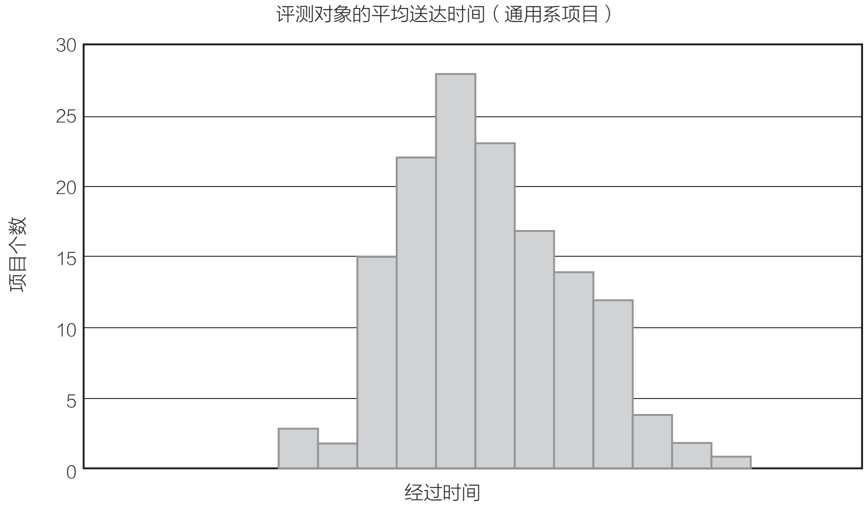
根据开发成果的最终更新日期推测缺陷

从图7可以看出，在各阶段收尾时，成果还有更新。因此可以推断，最终成果很可能缺少质量评测和自检。另外，判定是否残留有未定事项也是缺陷检查的内容之一。

▼表1　指标－征兆－假设定义例1

项　目	内　　容
指标	测定目标资料送达时间
定义	宣称开始评测到资料实际送达为止的时间间隔
基准值（例）	3天以上（具体情况基于组织、体制及项目规模）
征兆	PM管理层宣布实施评测后，资料提交发生延迟或花费的时间
假设	•在宣布实施评测后才制作了最新版（流程管理不善） •在宣布实施评测后才把最新版发布到服务器上（质量管理不善） •提交的目标资料不全（文件个数、代码文件数等）（预想和实际成果的对比管理不善） •质量负责人或相关责任者不在（组织体制管理不善） •没有规定从服务器提出、备份的操作步骤（运用流程管理不善）
预想缺陷	•成果之间逻辑不匹配（作者间不匹配、制作时间逻辑不匹配） •错字漏字、格式等轻微缺陷（这在实际检查中通常是无视或警告） •版本间不匹配 •数据/处理等接口不匹配

▼图6 评测对象的平均送达时间(通用系项目)



▼表2 指标-征兆-假设定义例2

项 目	内 容
指标	成果的最后更新日期和最后更新时间
定义	最后更新成果的日期/时间
基准值(例)	无(为了进行全体倾向分析和反模式识别)
征兆	共计16种流程反模式
假设	确认流程中是否存在反模式缺陷
预想缺陷	•深夜工作带来的缺陷=成果之间的不匹配(起因是交流问题) •从要求式样到设计式样缺乏可追踪性(进度上产生了一定间隔) •阶段性收工时的通宵、深夜工作=改定时因疏忽而造成的错误(起因是检查不足) •阶段性收工时残留有未确定事项(起因是检查不足)

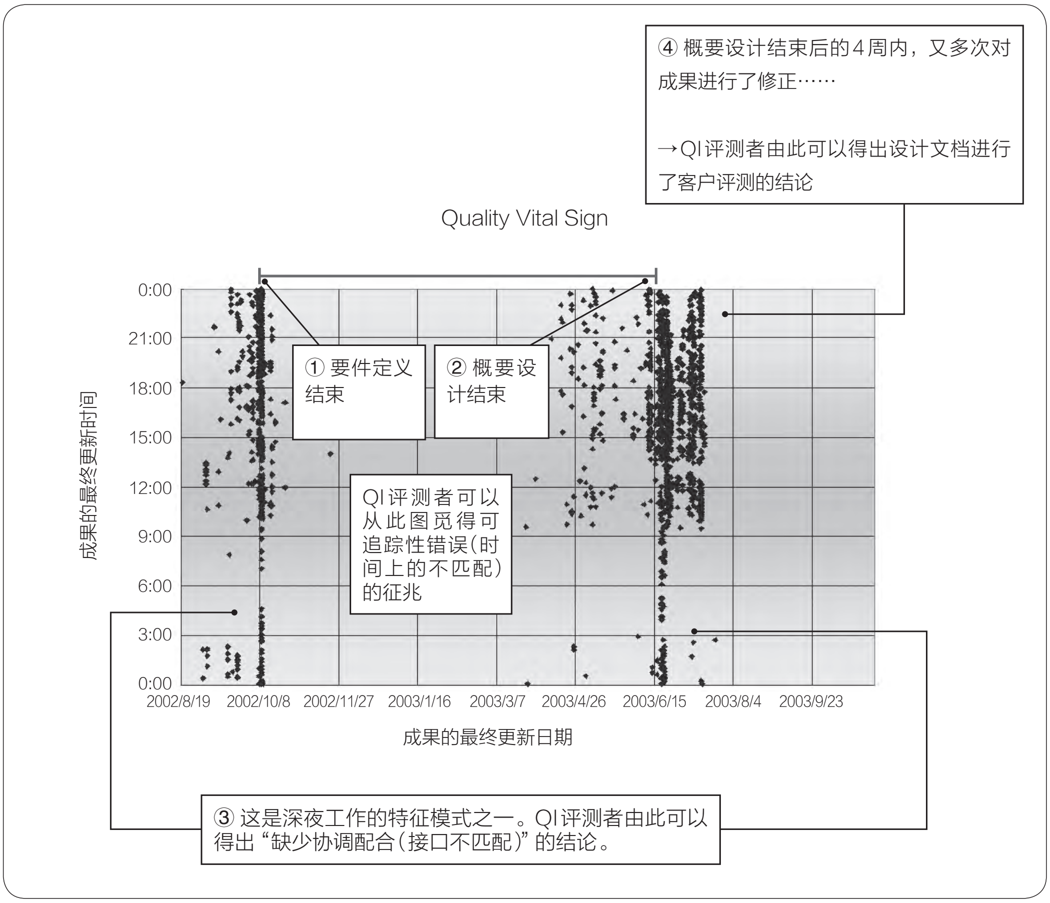
接下来，还可以从图中找到深夜工作的痕迹。由此可以推断，成果中很可能对式样内容的推敲、确认不够充分。因为深夜工作时，很多情况下都不进行确认检查，开发者各自沉浸在自己的规格说明书中，从而忽视了与他人成果的整合确认。最后，图7中左右两大块成果之间间隔了8个月都没有更新规格说明书，由此看来，项目很有可能缺乏可跟踪性。

从图把握项目，关键在于活用成果的“最终更新日期”“最终更新时间”等非质量指标。

缺陷密度、测试密度等经典质量指标，测定负荷较大，大部分是用来揭示“缺陷”。这些指标也称为“事后测定指标”，即作业结束后进行测定的指标。不过，在缺陷产生后，即使再用指标来评测也不能改变什么了。

质量评测中前置的倾向性评测指标，若不能在时间序列变化中检测出缺陷的话，就没有多大的效果了。图7中最终更新日期的例子，正是因为可以随着项目进程测定每天的数据，从而才能够得到“模式出现的征兆”。

▼图7　成果物的最终更新日期+最终更新时间



▼表3　根据关键字和集合图做出假设

项　目	内　　容
指标	代码中的单词统计+基础指标
定义	统计代码中关键字，预测潜在缺陷
基准值	无(基于所分析代码的语言、项目的人数、项目的规模)
征兆	存在与平均值偏差很大的事例
假设	参照图8

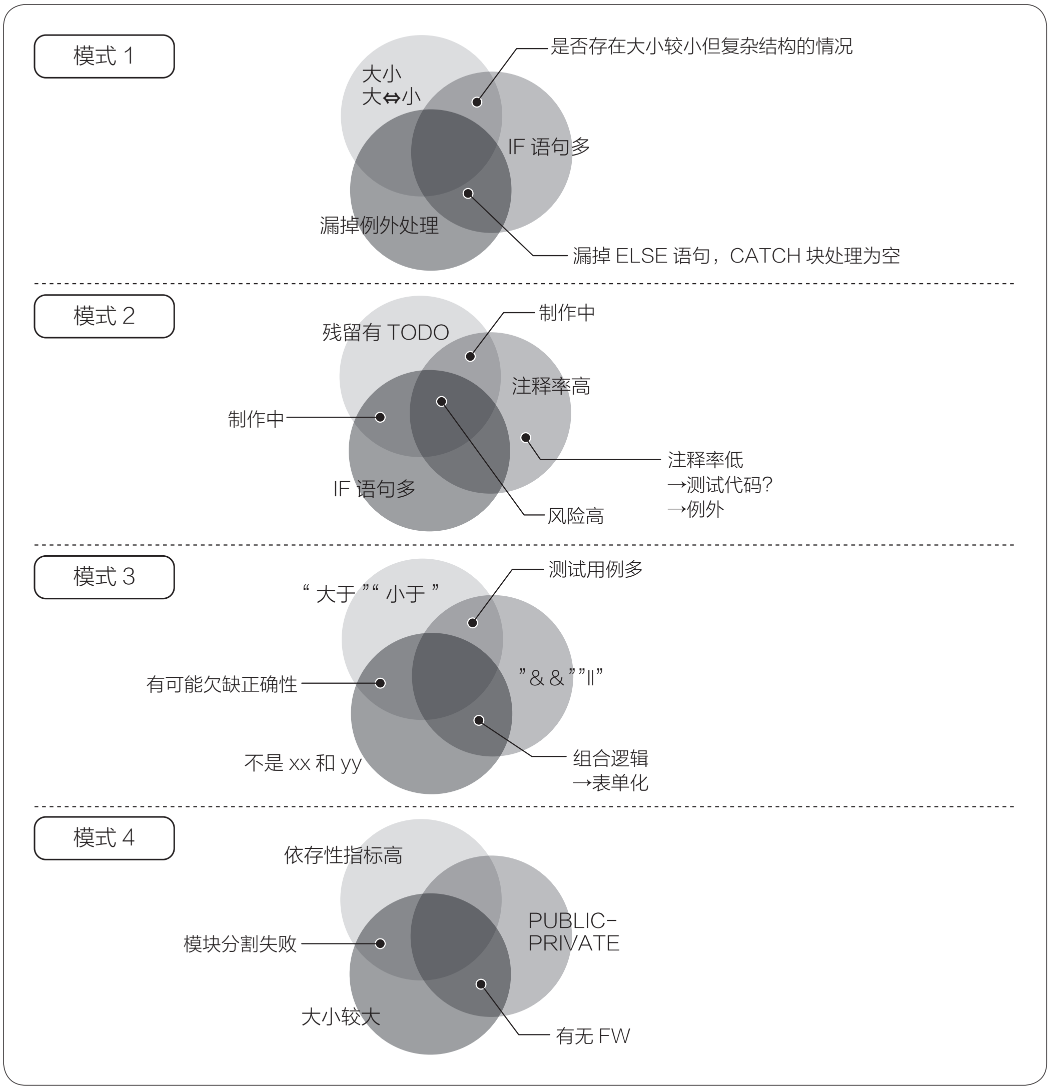
各位的组织、项目中都会出现何种模式呢?请一定做一下调查。

从征兆到假设的思考

接下来就要对缺陷或质量整体倾向做出假设。以假定有大量源代码为例，这里介绍一下质量检查之前做出假设的思考方法。

请看图8，它将表3测定数据的理解方式、与假设的潜在关联、检索方式等用集合图表示出来了。为了从大量代码中选择性地抽出缺陷率高的部分，可以通过组合使用几种模式，从

▼图8　从指标做出假设，然后锁定对象的思考方法（用集合图表示）



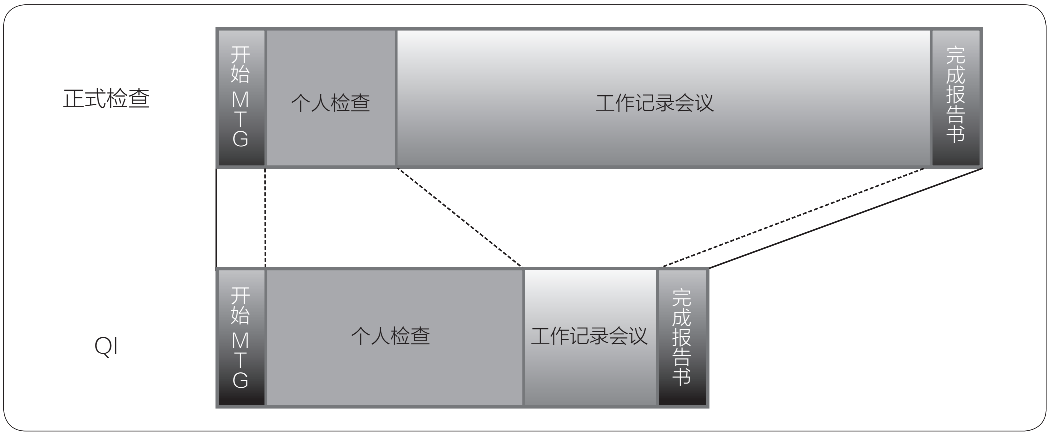
而使检查更快、更有效。比如“模式1”是这样的一种思考方法：它由代码的行数、IF分支条件个数、例外处理（比如Java的Try和Catch块的个数）组成，抽取具有如下特征的事例：代码行数少，但复杂的类需要增加单体测试用例，全面测试分支处理的用例设计很复杂。

“模式2”可以将具有以下特征的事例作为候补抽取出来：复杂的类正在制作且规格中未确定、因测试用例不足而隐含风险（这是因为未确定、需要讨论的功能+复杂度高+注释每次发生改变都被记录下来了）。

“模式3”和“模式4”是抽取如下事例的基准：设计中残存的含糊不清之处、功能不完善、违反标准、标准本身不完善等因流程不完备导致的缺陷。

实际也可以用表计算工具的过滤器锁定对

▼图9　缩短评测时间的高效个人检查法



象，但是检测模式用集合图（文氏图）将其画出来并保存是很重要的环节。利用已有的集合图，采用如下思维进行记录。

指标→征兆→假设→选择检查对象

如果可以实现指标数据测定的自动化，即使欠缺经验的人，也可以利用模式集以较高的准确率找出错误。笔者的经历即可证实这一点。

实际情况中，测定指标的数据及征兆、做出的假设、实际缺陷率等，会记录为“一套”数据。这是模仿医学病例的写法，优点是可以对应更复杂的思考方式。

事先了解一些经典模式，可以缩小检查范围。这能使质量评测活动的效率和效果发生飞跃性提升。

用集合图保存质量的方法——正如医生记录病例一样——可以实现思考过程/方法的共享与传达，从而成为很有价值的企业财产。

个人检查揭示出的缺陷

做出假设后，应该实施个人检查，进行重点性集中评测，只将重大缺陷递交到工作记录会议，从而减轻评测负荷。

正式检查与QI的区别如图9所示。

一般评测会议（这里称作工作记录会议）的负荷会大幅降低，从而也减轻了整体评测的负荷。这时候，要核对各自检查出的缺陷列表。合集评测整体指出的地方（逻辑与）以及评测个体指出的地方（逻辑异或），并从结果分析整体倾向，明确行动计划列表的不足，以及需要调查的重点（从分配检查部分与检查不足部分推导出重点）。

在这之后，根据需要进行再检查或追加检查。虽以预订时间剩余的15%为上限进行追加或再检查，但已经检查出的不均衡缺陷是否仅存在于局部（是否可能是整体共通的缺陷）？能否在其他地方验证已经检查出的重大缺陷？这些问题也需要进行交叉检查。

应该注意的是，在其他地方验证已经检查出的重大缺陷——这是再检查的出发点，而不能以缺陷数量的多少为评判标准。

举行工作记录会议“围剿”缺陷

工作记录会议，是以集合形式核对检出缺

陷的交流会。这里也用分类计数时使用的定量数据，对检查出的缺陷进行定性倾向分析。例如，假设有4个评测员，所有人都注意到的缺陷作为一个集合称“逻辑与”（AND），只有一个人注意到的缺陷称“逻辑异或”（XOR）。通常情况下，AND缺陷的检出难度较低。

假设开发方进行了检查（通称自检）后还有AND缺陷残留，那么其自检可以基本视为无效。反之，在检出很多XOR缺陷时，就表示可能还残留有很多检查难度较高的缺陷。此时，就有必要研究是否应延长评测时间，以补充质量评测。

通过工作记录会议核对缺陷，可以推敲出整体倾向，从而为最后的“反馈会议”做好准备。

具体的准备是指，要进行如图10所示的缺陷（=指摘缺陷）确认。

举行反馈会议

评测活动，是将检查出的缺陷传达给开发者，以促进项目改善。因此，比起把大量的缺陷列表返回给开发者，更应该根据实情返回一些比较实际的缺陷。

不过，日本国内各企业的质量评测活动，多数是只返回缺陷列表。究其原因，就是没能很好地举行反馈会议。图11所示是反馈会议前必须确认的4点事项。

即使检查出重要缺陷，若开发方不去解决缺陷，结果也毫无意义。指出众多无针对性或无实际意义的缺陷是浪费时间。测定出的缺陷的可修正数（有时指缺陷个数，有时指缺陷种类的个数），决定了测定评测活动的价值，同时也可以验证假设和前期指标数据测定的有效程度，这些都可以成为反馈的原材料。

如上所示，做出假设，以最低限度的指标数据分析征兆，就可以基于假设锁定缺陷的位置和种类。这种方法可以应用于所有领域（比如测试和项目管理的评测等），从而能在保证评测质量的前提下提高效率。

另外，与一般的评测方法相比，QI法能以1/16的时间达到几乎相同效果的调查结果（图12）。

▼图10　查出缺陷核对表

核对表例

- ☐ 确定检测缺陷的优先次序
- ☐ 确认在检测缺陷时，是否存在障碍性制约事项
- ☐ 确认在检测缺陷时，是否存在障碍性对外契约条目
- ☐ 研究缺陷检测的实施顺序（确认依存关系／是否有诱发其他缺陷的风险等）
- ☐ 是否考虑到计数错误
- ☐ 是否存在错误检出的可能性
- ☐ 明确反馈对象与反馈时间

眼力③ 单体/个人的评测技术

QI法的核心是从大量对象资料中抽取出目标对象。实际上，也有很多单体对象的评测技术。不局限于数据，通过仔细观察，很多情况下可以有效缩短质量评测活动的时间。

从尾部开始读代码

最费时费力的就是代码检查。“选取重点”是比较流行检查方法。一般来讲，也有人会把代码用编辑器打开，从第一行开始认真地边记边读，但这样做太花费时间了。

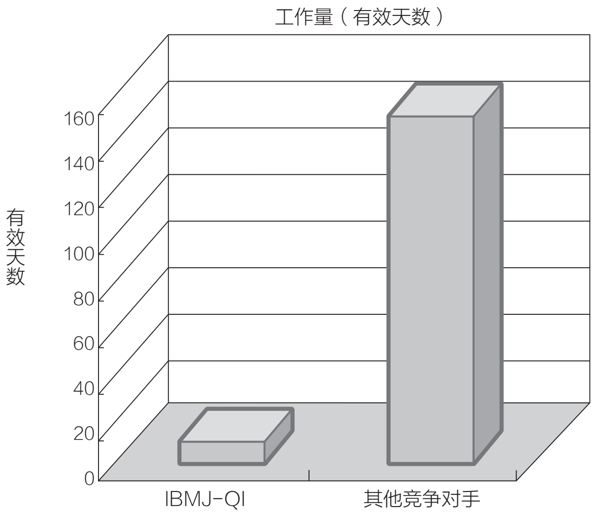
那么应该如何检查代码呢？实际上，可以

▼图11　召开反馈会议前的核对表

核对表例

- ☐ 反馈会议参会人员是否正确
- ☐ 反馈会议的目的是否明确
- ☐ 反馈检查出的缺陷能否用一句话说明白（缺陷的说明是否过于冗长）
- ☐ 反馈缺陷检出有无事实根据（是否包含推测的成分）

▼图12　QI法的生产性和效果



A) 工作量所需天数
- 其他竞争对手　150个工作日
- IBM-Japan QI小组　9个工作日

B) 可检查出的缺陷类型
- 一般评测法　可检查出64种缺陷
- Quality Inspection　可检查出62种缺陷

不用选取重点的方法。如图13所示，可以利用“最后修改的地方一般是末尾”的性质。因此，检查代码要从最后一行逆序进行解读。此时要寻找已有代码和新增部分中的变量名和函数名命名规则发生变化的“边界”。

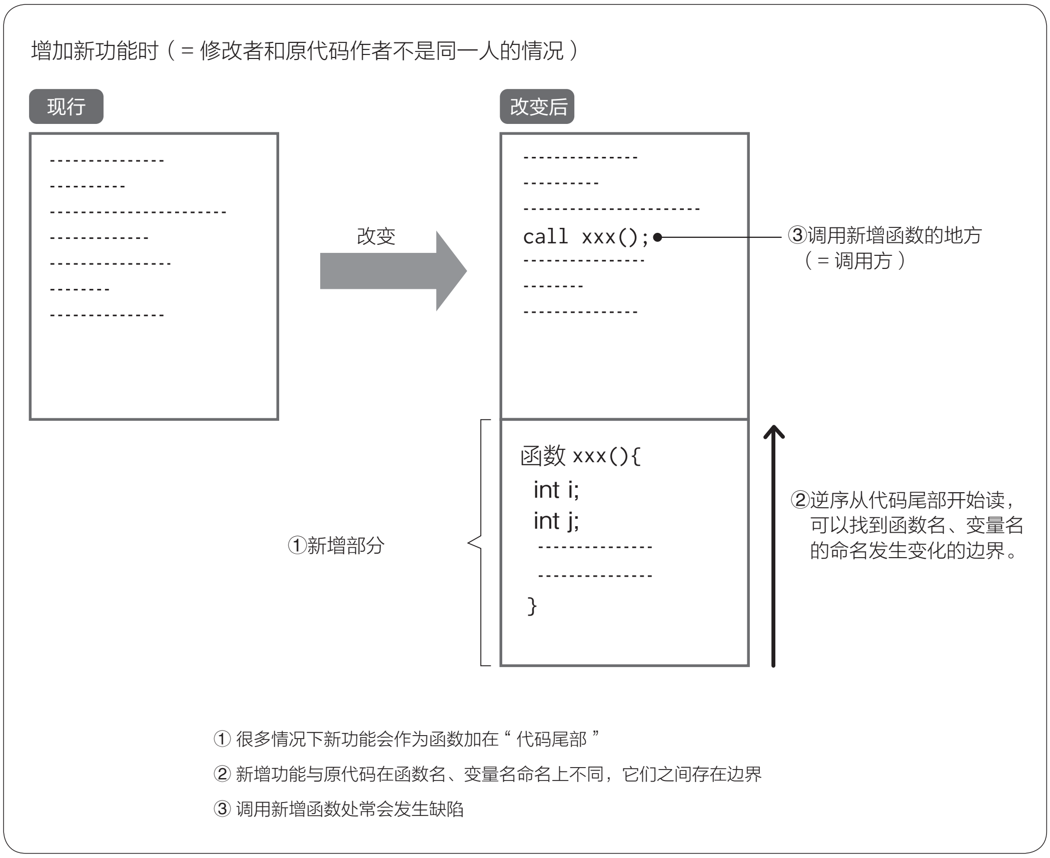
假设边界往后是新增部分，那么就要在已有代码中寻找调用新增部分的地方。因为在这些地方及附近存在缺陷的可能性很高（若不能很好地理解调用新增部分的前后及附近的已有代码，就不能进行调用）。利用“新增在代码的末尾的性质”，在很多情况下就不需重读所有的代码，尤其是在功能扩充或派生时，这种方法很有效。

图形化纵观代码

把代码复制到Microsoft Word等，调整显示比例可以看到文件整体。这样种形式易于纵观控制结构，从而便于找到代码检查的“突破口”。纵观重点如图14所示。

通过调整显示比例，模糊化对象代码，使其成为单纯的“图形”。从宏观角度观察图形，随着显示比例减小，一些地方好似被涂黑了，通常这意味着这些地方的代码处理比较多。缩进的地方表示分支、循环语句较多。这样，代码中的重点难点就可以一目了然了。而且，还可以从图中看到同样的模块（以“同样的形状”）多次重复出

▼图13　眼力技术之代码尾读法：修改或追加时从尾部开始读



现，这实际上是代码的复制/粘贴引起的。

这种方法，无需从头开始读代码，而是从视觉上直观把握代码整体构造，从而明确评测重点。习惯以后，就能熟练根据“形状”来判断代码的潜在问题。

对代码排序（观察程序代码密度）

把代码各行的字符数降序排列，可以检查出是否存在复制/粘帖，是否存在遗漏处理等。

代码排序结果可以得出，①复制数行代码的痕迹，②在发生条件分支时注释被遗漏的可能性，③卷值是否正确、是否未注明等，基于上述结果，就可以在评测开始时锁定评测候补目标。根据不同情况，有时还能发现漏掉式样的地方。

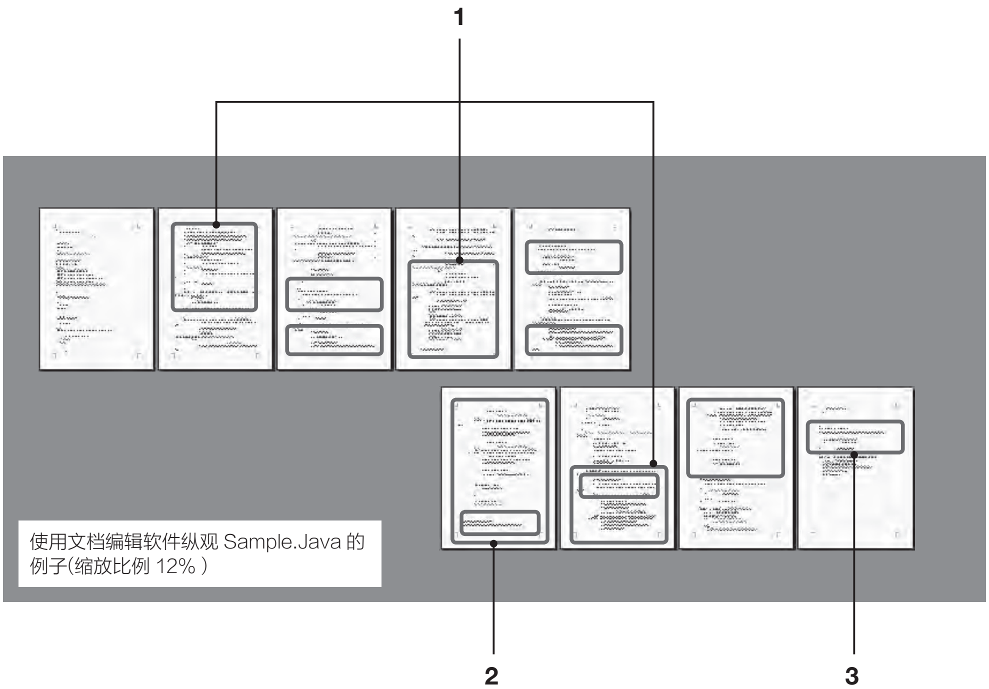
以“块”的形式表示IF语句、For语句等主要控制关键字，这便于掌握网罗性、变量名的命名规则错误、局部变量错误等，并由此得出错误代码进行修改方案。

◆　　◆　　◆

这次所讲的技术，都是“模糊地”纵观对象整体，进而关注对象的“局部细节”。不过技术不只限于一种，应该把多种技术组合使用。

不管是哪种方法，都要填写Bug表，或者

▼图14　眼力技术“图形化纵观代码”



检查重点:
1. 黑色的地方 = 处理较复杂，可能是比较重要的部分
2. 向右凹陷(缩进)的部分 = 可能是多层嵌套、构造复杂的分支或循环
3. 能看到方法块吗？ = 检查方法注释和方法的统一性

将Bug信息进行整理。虽说可能不足以作为调试（发出调试指示）信息，但可以大幅缩短评测中检查出缺陷的时间，而且效果也能得到保障。这也就是所谓的“实践智慧=实用技巧”吧。

读者中可能会对这些技术有“原来如此”的共鸣吧。这些方法并不能作为理论确立起来，但现实中还是很有成效的。

为何这种非理论的实践技术很有效呢？实际上答案在人类获得关于质量评测活动、特别是检查出缺陷能力的层面之中。

检查出缺陷的能力——感觉和认识

在测试或评测中，查缺陷的能力一般被认为仅是寻找错误的能力。也有人认为只要多学

▼图15　对代码排序（例：按字符数排序）

```
/**
 * 用户注册与登录系统
 */
public class userRegister implements ActionListener {
          ArrayList<User> list = new ArrayList<User>();
          // 用户名输入框
          JTextField jtf1 = new JTextField(14);
          // 密码输入框
          JTextField jtf2 = new JTextField(14);
          // 邮箱地址输入框
          JTextField jtf3 = new JTextField(14);
          // 重新输入密码输入框
          JTextField jtf4 = new JTextField(14);
          JFrame jf;

          /*
           * 界面事件的处理
           * 用户点击注册时，检查用户的填写信息，将用户注册信息写入文件
           * 用户点击登录时，若用户已经存在时，显示登陆成功，否则登录失败
           */
                    .
                    .
                    .
```

按字符数排序后

```
writeToFile();
while (true) {
user.setUserName(jtf1.getText());
user.setPassWord(jtf2.getText());
user.setEmail(jtf4.getText());
User user = new User();
User temp = new User(jtf1.getText(), jtf2.getText(), "q");
User read = null;
try {
try {
try {
this.userName = userName;
this.setUserName(userName);
this.setPassWord(passWord);
this.setEmail(email);
this.passWord = passWord;
this.email = email;
System.exit(0);
super();
String str = e.getActionCommand();
                    .
                    .
                    .
```

习一些检查出缺陷的方法，检查出缺陷的能力就会相应提高，进而成为优秀的测试或评测人员。

但事实并非如此。检查出缺陷能力由以下2种能力构成（图16、图17）。

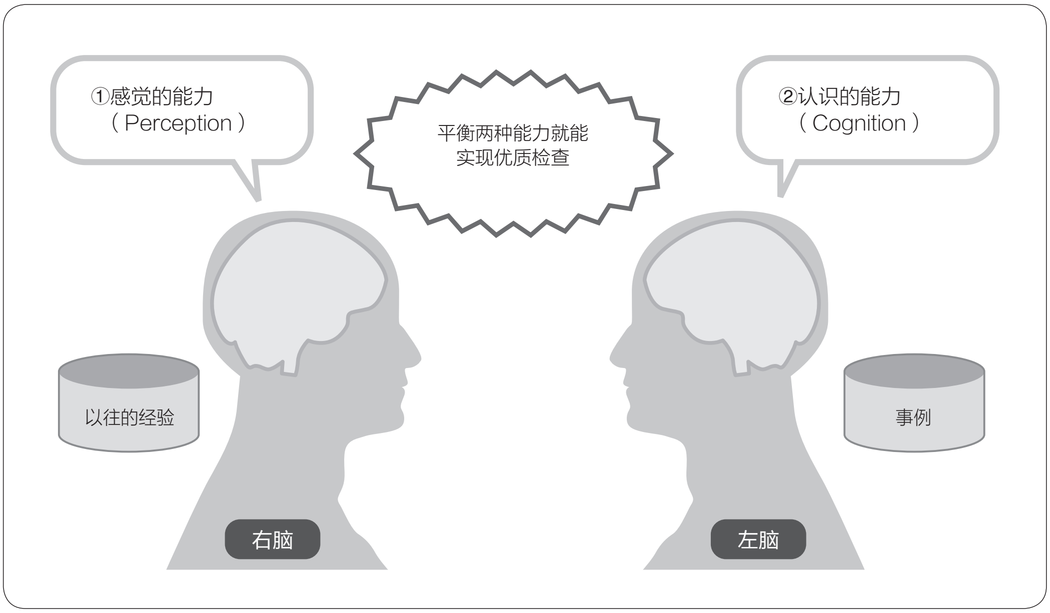
① 感觉的能力（Perception）

即观察（模糊地纵观整体）、洞察（关注一定程度的细节）等能力，通过系统地检查，直观地感觉缺陷的潜在位置，并根据经验、推测以及参考以往例子进行验证。

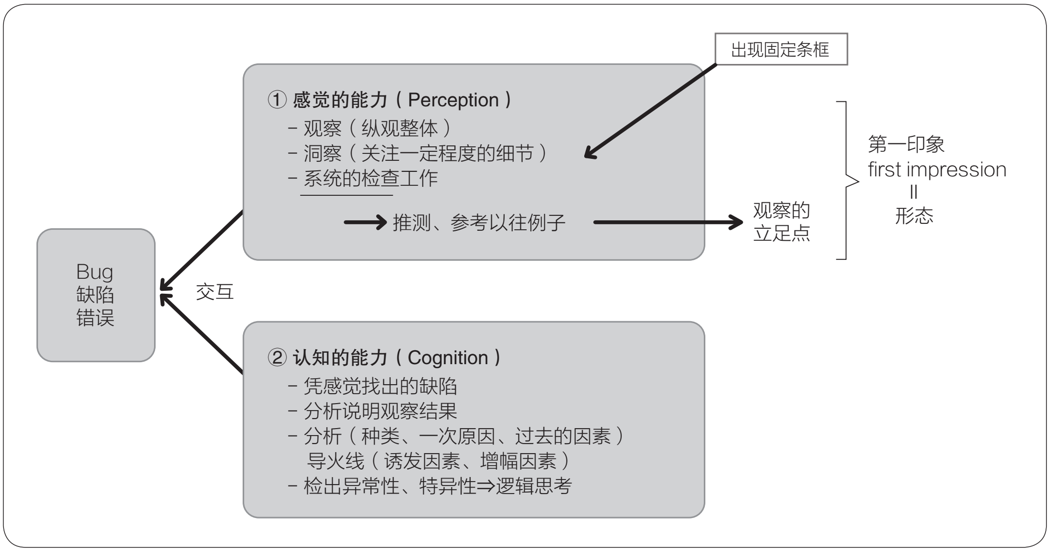
② 认识的能力（ Cognition）

是指对凭感觉出的缺陷，进行确认等分析工作（种类、位置）。特别是检查出异常性、特

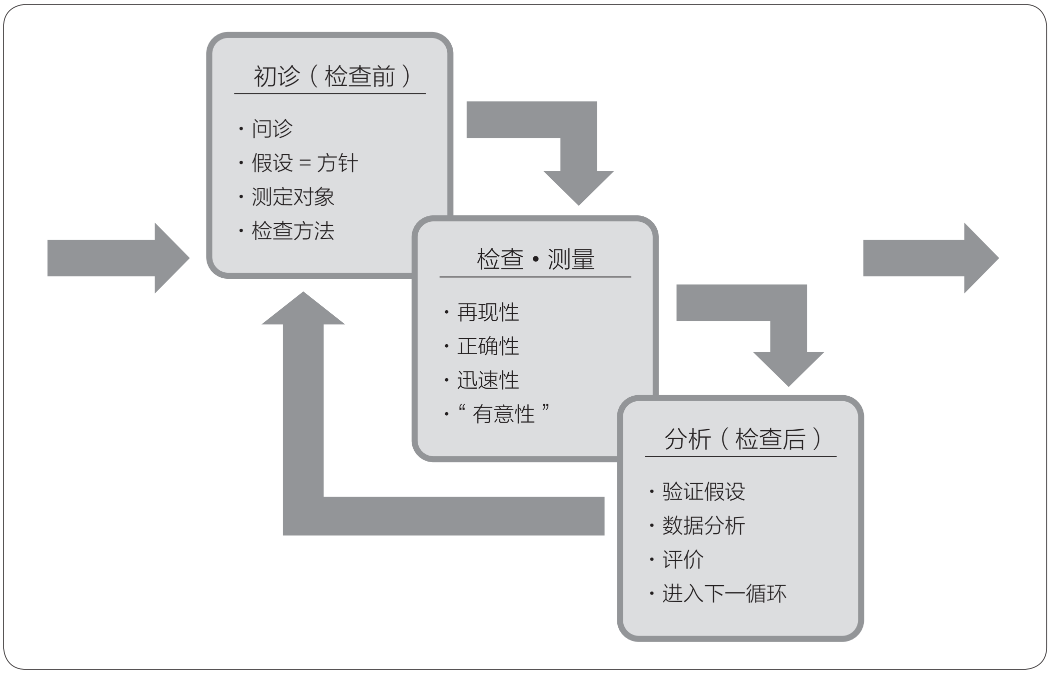
▼图16　检出缺陷的两种能力“感觉和认识”



▼图17　感觉和认识的定义



▼图18　医学的“诊断”循环



异性时，通过逻辑思考“能够理解和说明所存在的缺陷”。

像②这样强化理性、逻辑思考的能力，是对检查出的缺陷进行分类和整理，只要花时间学习，是比较容易掌握的能力（积累Bug笔记，记录每天出现的缺陷，和他人进行缺陷交流也是某种程度上的学习）。

发现错误的能力，也就是①的感觉能力，必须进过特别训练才能提高。本文的主题“眼力”，实际指的就是这种能力。

通常，①这种难以掌握的感觉能力，可以通过使用定量数据的方法来弥补。

把握征兆——结合医疗领域的定量数据测定

在医学界，医生为了确定症状要进行问诊和观察。根据问诊获取得信息，从已知的无数病症中锁定2至3个作为候补病症。

医生采用的“排除法”思考方式，也就是验证信息与征兆的对应关系，通过这样的排除，可以从大量的疾病中找出准确的疾病（像这样结合特征征兆和实际原因、病理，并以医生经验作为补充的体系称为征候学）。确定2至3种疾病候补的过程就是“做出假设”（图18）。

做出假设也就是所谓的“诊断”，医生就可以根据经验以及以往的病例进行推测。为了“确定”假设，需取得检查数据以进行验证。医生的“眼力”，相当于前面提到的“感觉”（从征兆开始到做出假设的过程），以及进行数据验证等一连串思考。要说这就是作为医生应有的能力也不为过。

这种模式，在软件开发领域又是怎样的呢？

毕竟不能对电脑问诊说“老兄，有什么样的症状？”所以需要立足于取得优质数据，观察其征兆，然后根据状况进行推测，如果是相同

状况或是以前出现过的状况，就将之与模式进行对照。接着把从数据征兆、做出的假设、实际的缺陷进行关联记忆。

这种关联记忆征兆数据和假设数据的方法，是软件质量评测专家和大部分开发人员提高技术能力的有效手段。

简化测定指标

世间存在一种古典、固定的“经典指标”。诸位读者是否对这种指标有所了解呢？

指标能表示复杂度、大小等，不过不少人认为指标测定的负荷都较大，必须收集一定数量的标本才可以。

用于质量验证的指标，规格无需过高，简单的东西就可以。质量检查，只要找到关于检查启示、契机、特定缺陷的“立足点”，不管是什么都可以。

即使像本文事例中“文件的最终更新日期 · 时间”那样的简单指标，只要连续测定也是足够有用的。“足够有用”的感觉很重要。用一个很早以前开始使用的词来形容就是Good Enough，这是质量评测中最应该优先考虑的质量目标。从这个观点来看，能够全力投入质量活动并可进行快速测定的指标是最理想的。与追求质量评测活动的完全性相比，生产性和速度的理念显得更重要。

不要以一个测定指标的绝对值判断质量的好坏

大家知道循环复杂度（CC，Cyclomatic Complexity）指标吗？循环复杂度指标由Thomas McCabe设计发明，以分支数或调用数来计算程序内部的复杂度。这个指标在日本也得到了广泛的应用，将其作为既定测量指标的组织也不在少数。甚至还有这样的说法：“如果这个值超过30，就要对构造产生疑问。如果超过51，测试将会很困难。如果超过75，不管什么修改都可能成为误修改的原因。”不怕被误解地说一句，这样激进的管理可以称作是“指标的陈规陋习”，事实上此指标与缺陷、质量基本无关。

如果把精力都放到单一指标上，在不少情况下会忽略模块间的复杂度（IMC，Inter Module Complexity，模块间相互复杂度）。

模块间相互复杂度和循环复杂度如图19所示，是成相反关系的，所以使用者应注意两者的平衡。但现实中，在日本国内灵活运用这两者进行测定的例子很少见，或许是因为负荷太大吧。

事实上，组合使用小负荷测定方法对软件质量进行综合评价，要比使用单一指标更有用，同时也会成为前面所讲的“质量评价立足点”、“模式积累”等的重要资产。

比技术更重要的东西

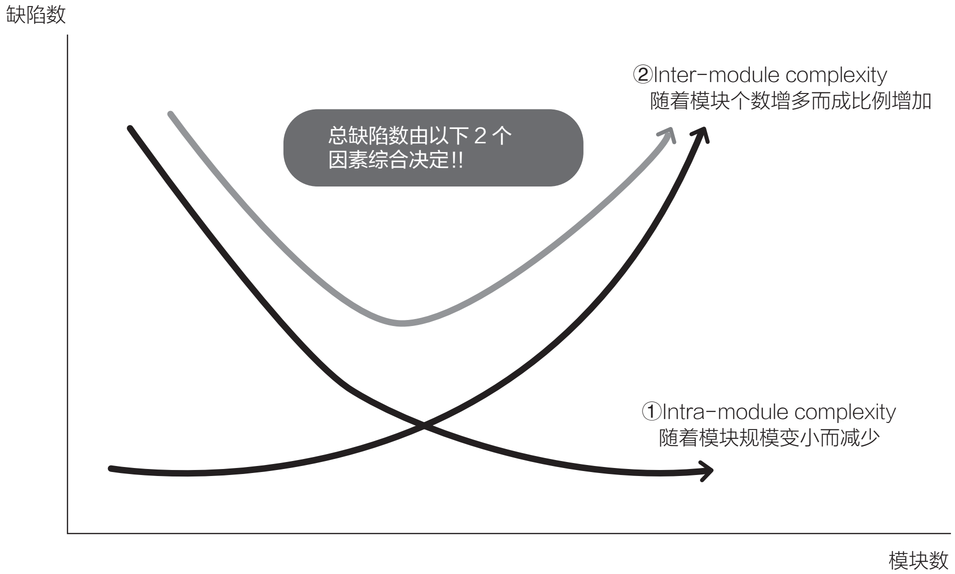
“着眼于对象”是与指标并列的一种“眼力”，这种思考方式甚至比技术更为重要。

Frederick P. Brooks曾在1980年用“没有狙击狼人的银弹”[3]的表现手法为IT界敲响了警钟——万能的软件开发技术是不存在的。

试想一下，假设大家的手中有“狙击狼人的银弹”这种魔法般的武器，大家在实际使用这个新武器之前，应该做怎样的准备呢？选什么样的枪？射击距离是多少？可否单手射击？还可能会讨论一些技术方面的问题，最后还可能会准备魔法用的咒语。

但是，最关键的是要了解对手。如果无法

▼图19　两种复杂度



① Intra-module complexity（例：Thomas McCabe's Cyclomatic Complexity）
=1个模块内的复杂度

② Inter-module complexity（例：Henry-Kafura's Inter-Module Complexity）
=和其他模块相互作用的复杂度

识别、区分对方是否是狼人，就可能把银弹射向无辜百姓。所以首先应该准备的就是“识别狼人”的方法，仔细观察对象是很有必要的。

那么，对于开发工程师、质量工程师、测试工程师而言，什么是“对象”呢？不用说，一定是“缺陷”了。关于缺陷的知识和技术交流不畅通，比如“资源死锁”“内存泄露”等知识，应该如何教给新员工呢？

不知是否有关于“正确”的定义，或者说关于“正确”原理的书籍和论文，如果有的话，如图20所示的3个状态，肯定都会被定为“缺陷”吧。

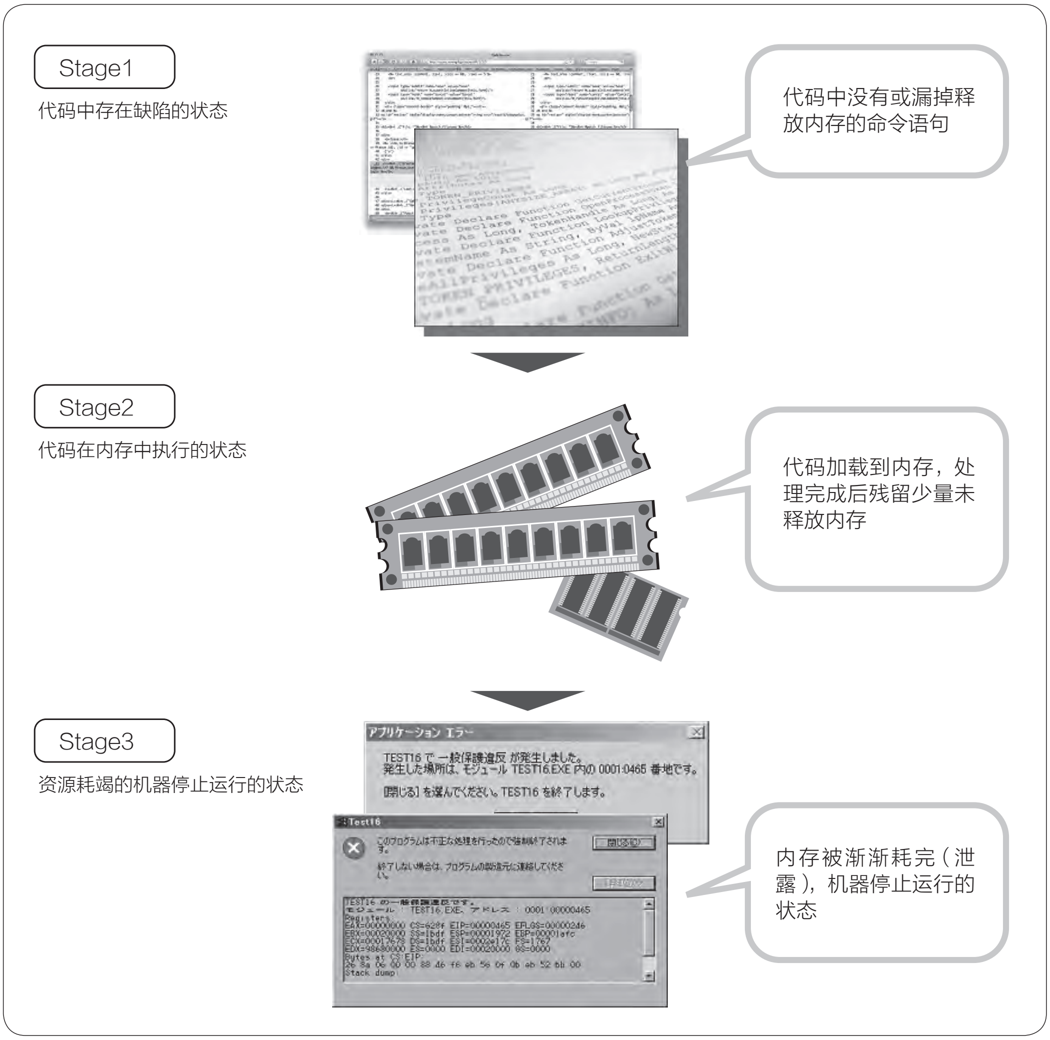
还有一个重要的问题。如何用指标来测定图中所示的3个缺陷呢？这些都属于1个（一种）缺陷吗？还是按3个来计算的？还是说有20个地方发生这样的错误就按20个来计算呢？

图20中所示的3种状态都是日常开发项目的测试、评测中频繁出现的，应该是诸位读者比较熟悉的缺陷。但是，因为缺少关于这些缺陷的正确定义和测定方法，所以这些缺陷经常就这么稀里糊涂地被使用。缺陷的研究和知识积累并没有公开化，看到某个缺陷，“啊，我知道。”这样想的人需要自己来回想与此缺陷相关的知识是从哪里来的。很多人恐怕都是从前辈或上司“口传心授”的形式获取这些隐形知识的吧。

缺陷数据库——提升“眼力”的能源库

IT界不存在软件缺陷数据库。这本身是很

▼图20　什么样的状态称为“缺陷”呢？



不可思议的事情，因为在产业发展的同时一定会产生公共数据库。比如医学界有收集各种疾病的病理学数据库，对烹饪或酒而言，编撰有标准的菜谱、酿酒秘方，并且现在仍然在进行更新。

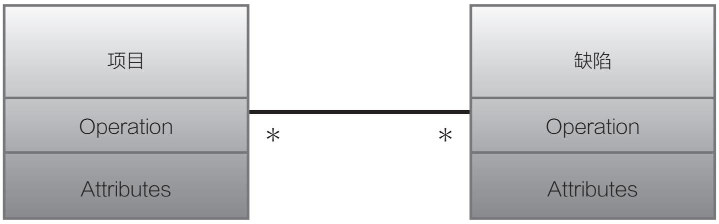
那么为什么不能把软件的缺陷信息积累起来，传承给下一代工程师呢？据笔者所知，确实也有这样做的企业，如“让新员工阅读大量以前的错误记录表”“全部记住BTS（Bug Tracking System）以往的记录”等。但因无法保证能引发对“缺陷缘何存在”的思考，所以这样的学习效果有限。而且，如图21所示，缺陷的“实例”是依赖于项目的，即使记住了也不能挪用到其他项目中。

图中的项目缺陷，也就是日常开发人员、测试人员要填写的“故障管理/错误记录表”，大多数情况都是按前辈口头传授写法，而并不参照缺陷库的信息（图21的②）进行填写。

▼图21 项目和缺陷以及项目缺陷之间的关系

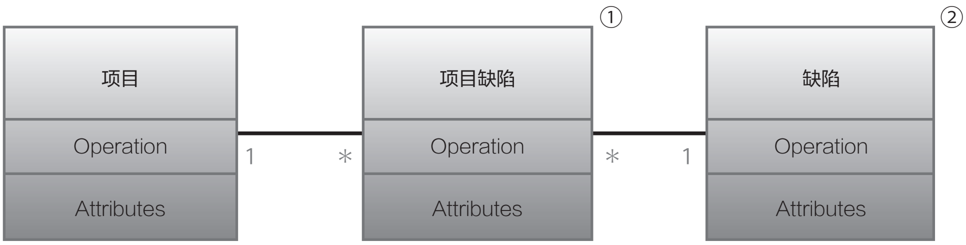
■ 项目与缺陷的关系如下所示

1. 项目有多个缺陷产生
2. 同一缺陷可以在多个项目中检测出来



■ 上述多对多的关系，可加一个1对1的关系“关联实体”进行整理

1. 在项目和缺陷间设定“项目缺陷”实体
2. 整理成1对多的关系



- 不管收集多少项目缺陷（①），也不能用于观察、参照、再利用。
- 想要的是②“缺陷库”信息。确定后、抽象化、整理后才能运用。

结果就是伴随着项目结束，这些信息也被很自然地删掉了。

假如能进行充分的抽象，去除对个别项目状况或技术的依赖性，制作出“缺陷库”的话，就可以预防同一类缺陷的再现，既不用再为检测技法的选择而烦恼，也不会发生因没有意识到副作用而选择了错误的排除法的事了。

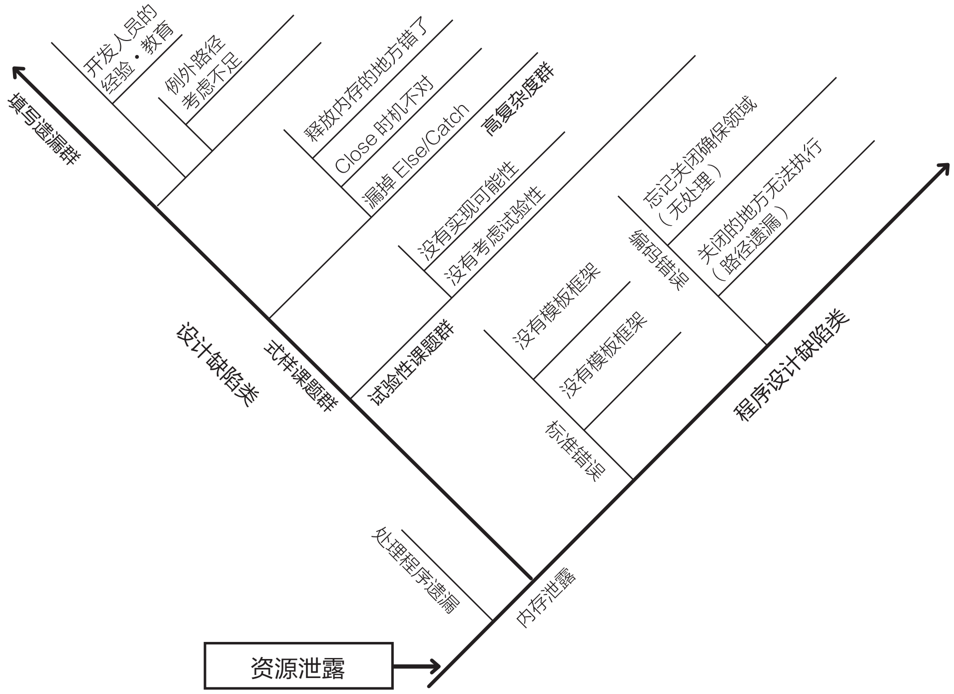
关于特定缺陷的定义、特性，检出方法，发现征兆、做出假设的方法，排除法、副作用，预防方法等，通过参照定义了这些内容的库数据，就可以排除人为原因的干扰。

比如对于内存泄露而言，可以像图22那样用“生物进化树”的形式总结混入原因。通过追寻关联缺陷“树干”可以使检查工作变得简单。

如果缺陷库可以得到普及的话，开发者和质量评测员的工作会发生什么样的变化呢？根据医学的发展进程，笔者认为可以进步到如图23所示的程度。

首先，现在有一部分评测员和测试员成为了专家，不少情况下也构筑成与开发方的对立关系。对于开发方而言，进行测试和评测就像日常生活中经常被唠叨要注意身体健康那样，觉得啰嗦又不得不做的事。因此，执行定量测定、缺陷跟踪和收集在层级和优先度上都高于缺陷库。

▼图22 树形图表示内存泄露的原因



解说：内存泄露

语言依存性：有
检查方法：动态测试
有效对策：在式样阶段预防（评测）
检查式样、代码在时间上的是否匹配
注意点・并发症：多余的关闭处理（＝释放了没有确保的领域）

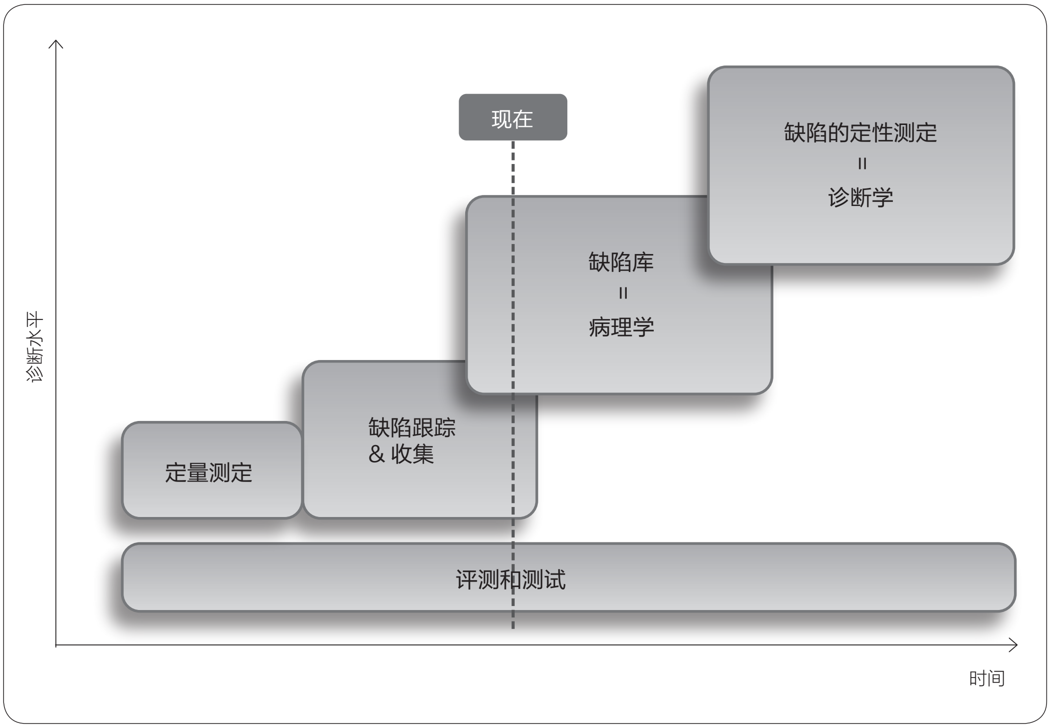
这也逐渐形成“缺陷知识='恶'的知识”的观点。而且，缺陷的定性测定、如同诊断医学，只需要把握少许征兆，就能促进处理方法、治疗方法甚至预防方法的高速发展。可以说高质量的技术评测时代即将到来。

未来时代，会提供各种各样的方法，以强化技术者的“眼力”，而缺陷数据的共享、传递、使用也将渐渐得以实现。这将是一个追求细微技术和方法论教育的时代，也意味着重视“对象=学习缺陷，学习'恶'的知识”和培养辨识缺陷“眼力”的时代已经不远了。

总结——银弹真的是必须的吗?

尽管如此，大家还是想追求部分最优化的技术或者是可以带来微小进步的“银弹”吗？追求产生差异化的技术（=银弹）不仅需要劳力和

▼图23　医学进步和IT产业技术进步

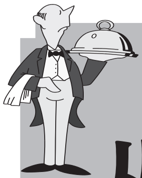


时间，笔者认为这会使本应观察现场、现实的目光变得模糊。如果这样的话，今后为了成为优秀的开发者，就需要有新的思考方法。需要锻炼的不仅是技能还有“眼力”，也就是可以看透事物的真正价值、从指标数据捕捉征兆、缺陷知识的“观察”技能，在未来的时代，没有这种“眼力”就不能生存。如果不能从这种新视角对自己开发的成果进行自我质量评测，在新的时代中就会成为“只会开发的开发者”而被淘汰。

不用说，代表各种各样开发技术、质量保证测试、评测的各个技法都是在漫长历史中积累出的产物，预先了解一下当然是有益的。但我们技术者最应该优先掌握的不是技术知识，而是缺陷知识和洞察缺陷的“眼力”。

本文通过纵观质量工程师、测试工程师平常使用的各种质量评价技术，解说了今后开发系的技术者应该具备的洞察系统的第三种力量“眼力”。笔者衷心地希望本文中“在质量中带入速度概念”的思想和所讲的“眼力”能够成为下一个新潮流的基石。

●参考文献
[1][2]　基尼系数：http://ja.wikipedia.org/wiki/基尼系数
[3]　《人月神话》(日文版)，Alfred Brooks J.(著)、泷泽徹、牧野祐子、富泽升(译)，2010年12月，ISBN：978-4864010054



按部就班真的可行吗?

从小规模工程学习活用Jenkins



第2回 有效管理、轻松运用

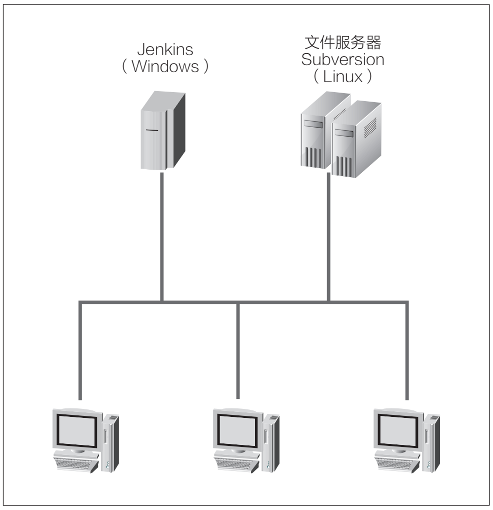
上期向大家介绍了笔者公司中Jenkins的使用情况，并直接对JOB进行了说明，而跳过了介绍Jenkins环境的环节。本期，笔者将介绍Jenkins环境的设置及管理。

文/岛崎聪(株)XVI　Twitter @sato_c　译/阿逸

Jenkins的运行环境

笔者的开发环境是Windows平台的Jenkins（图1）。起初，因为文件服务器和Subversion都是在Linux环境下运行，所以也想使用Linux下的Jenkins。但是图像格式转换、文件打包等作业中需要用到的工具只能在Windows上运行，因此选用了Windows平台的Jenkins。

●图1　项目中使用的网络结构



Jenkins中，由一台主服务器（master）通过网络连接多台从属PC（slave）共同执行JOB。原本计划使用Linux主机及Windows从属机的结构，但考虑到构建这样的结构非常繁琐，故本次只选用1台Windows主机为例进行说明。

因为只有1台主机，所以有一些需要注意的事项。笔者将以此作为本文的第1主题，在后文中进行详细介绍。

使用过程中出现的问题

实际操作中会遇到如下问题。

- JOB的记载方法
- 如何传递JOB的执行结果

一般我们可以通过浏览器轻松管理JOB的内容。然而，通过浏览器编辑JOB的记载内容却并不那么容易。本文的第2主题将介绍如何改善上述问题。

如何查看JOB运行的结果？确认Jenkins的

JOB列表画面不就行了吗？事情可没那么简单。肯定会有人觉得确认JOB列表画面很麻烦。就算告知Jenkins的JOB列表画面的URL，也会有人不知道这个JOB列表怎么看，以及运行的结果在哪里。如何让其他开发人员正确确认JOB的运行结果，将作为本文第3主题进行介绍。

①单主机的使用要点 ——工作区(workspace)的管理

首先，介绍构建单主机Jenkins环境时遇到的问题及其解决方案。

Jenkins运行JOB过程中需要的磁盘空间称为工作区。工作区是JOB的工作区域目录，运行JOB时，从仓库中获取新文件，保存至工作区后再进行处理。

工作区容量并非总是由存储内容决定。但仅对编译程序而言，存储下载的原编码以及保存目标文件等中间文件，1GB左右的容量就足够了。

不过，笔者还需要处理两类影像数据（用于目标硬件的影像和用于AfterEffects等创作工具的影像），仅此就需要占用大量的磁盘空间。虽说只要在Jenkins中使用大量硬盘，此类问题就能迎刃而解，但现实往往并非如此。这里，笔者为了节约磁盘容量，减少了每个JOB的数据量。

需要为每一个JOB准备工作区吗?

Jenkins的默认设置是每个JOB拥有独立的工作区。若想只使用1个工作区，就需要分析独立工作区和共用工作区各自的优缺点（图2a、图2b）。

● 独立工作区

若JOB使用独立工作区，其运行后，会从仓库中下载文件并保存至工作区。此时就需要保证工作区有足够的磁盘空间来存储源代码、数据以及编译过程中产生的中间文件。

JOB生成的目标文件和数据只要不提交到仓库就不会对其他的JOB产生影响。因为每个JOB有自己独立的磁盘空间，所以多个JOB同时运行也不会相互干扰。

使用其他JOB的输出作为输入的话，需要将输出保存到自己的工作区中。

● 共用工作区

多个JOB共用1个工作区，只需要1个工作区的磁盘容量，所以能很大程度地节约磁盘空间。

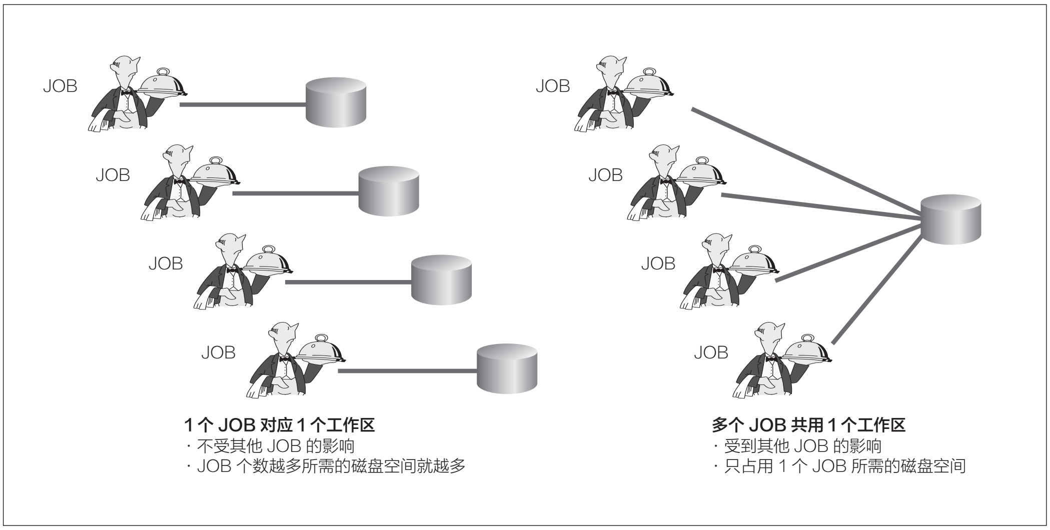
公用工作区的JOB同时运行，在同一磁盘空间中更新仓库文件，会造成更新冲突。运气好的话，冲突的双方报错并终止运行，但大多数情况下一方报错终止运行，另一方则一直保持运行状态而无法结束。如果JOB未正常结束，

Column 1 使用从属机(slave)

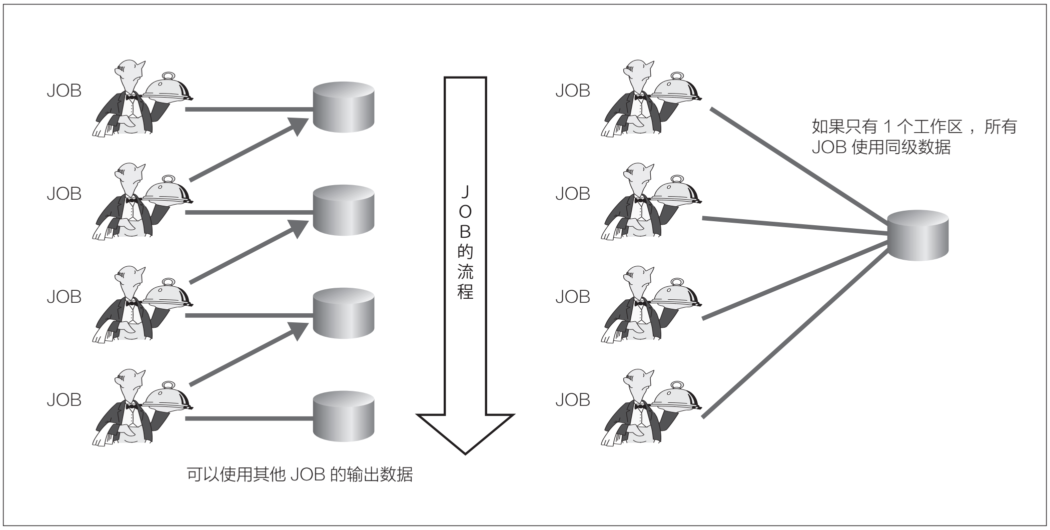
使用1台主机(master)进行作业不耗费太多时间的话，就没有必要使用从属机。当使用1台主机运行JOB耗时过长时，或者明确需要使用并列构建条件来构建多个编译时，需要使用从属机。没有明确从属机的职能就盲目使用，导致从属机闲置着，这是很大的浪费。

推荐使用从属机进行并行编译的例子有：编译英文版、中文版等不同语言版本的程序；编译iOS、Android等不同平台的APP。在从属机的多环境下，就可以使用对应iOS或Android的不同插件。

●图2a　每个JOB拥有独立的工作区（左）及共用1个工作区（右）的构成图



●图2b　每个JOB拥有独立的工作区（左）及共用1个工作区（右）的数据共享



下一次启动时，JOB会因为无法访问而无法正常运行。发生这样的问题是非常棘手的。

若共用1个工作区，输出、保存JOB结果的磁盘空间相同，就能直接使用，免去复制的麻烦。

以节约磁盘容量为目标

综合考虑了共用工作区的优缺点后，笔者决定本次以节约磁盘容量为主要目标。只让需要同时运行的JOB使用不同工作区，其他情况下所有的JOB共用1个工作区。

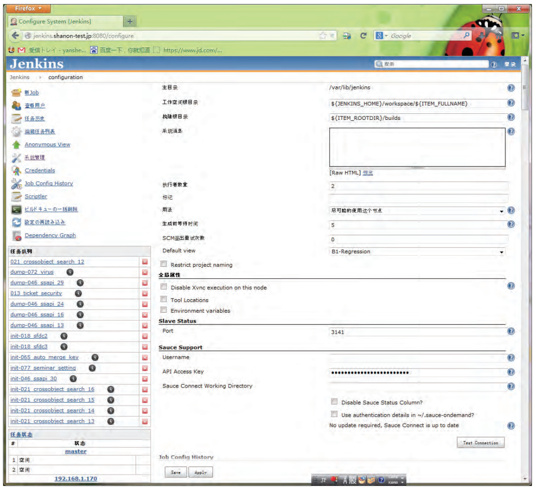
为了在JOB间共享工作区，需要对工作区进行自定义设置。通常，工作区位于Jenkins设置界面中指定的主目录下（图3）。自定义工作区的有效范围仅限于当前的JOB，因此需要在

JOB的设置界面中设置。一般情况下，自定义工作区的设置是不显示的，按下“高级项目选项”下的“高级设置”按键后，就能看见如图4中的设置项目了。

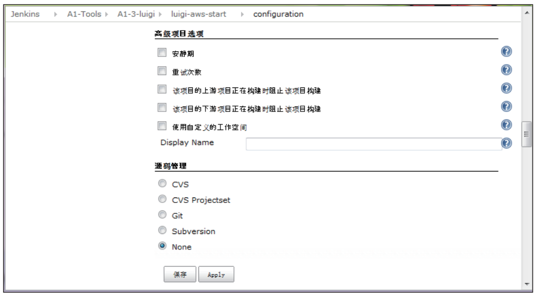
选中“使用自定义的工作空间”并指定目录，再运行JOB就会在新的路径下生成工作区，并以此作为工作路径。只需要对必要的JOB进行上述设置。

因此，这种简单的共用一个工作区的方法，虽然节省了磁盘空间，但是当几个JOB同时运行时，会产生数量上的限制。实际上，在本次开发中，除了视频的转换JOB能多任务并行运行之外，其他的都是单一任务。定期自动运行JOB也不会产生问题。

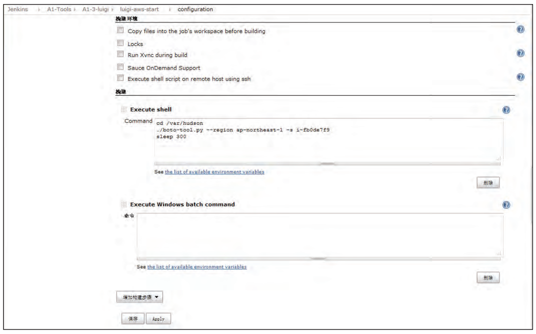
●图3　从系统设置中指定JOB的路径



●图4　设置自定义工作区



●图5　处理的编辑输入框



②JOB处理的记载

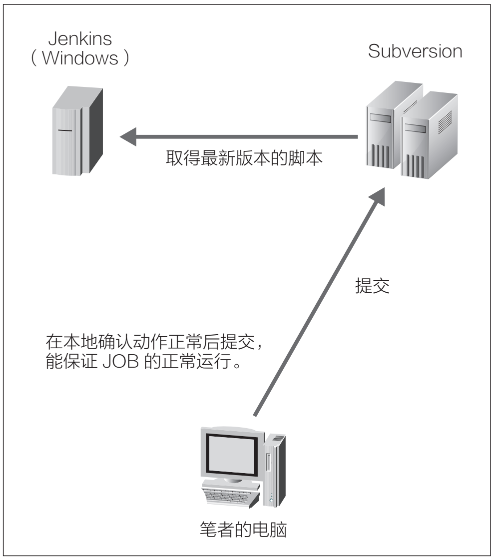
我们暂时解决了磁盘容量的问题，接着出现的问题是JOB处理的记载方法。JOB的处理可以记载在设置界面的构建处理中。但是，这样直接写在网页上的做法实在不怎么方便。如果是只有几行的处理那就算了，然而，用浏览器的输入框编辑较长的处理实在是一件很麻烦的事情(图5)。

除了简单测试之外，特别是在正式运用的阶段，应该尽可能地使用批处理文件或者shell脚本进行版本管理。JOB的生成以搭建Subversion等版本管理系统为前提，并会建立存放JOB处理脚本的文件夹进行统一管理。这样，就能将新的处理文件经常存放在Jenkins的工作区中了。

在用Jenkins运行脚本之前，先在本地对脚本进行动作确认，能够确保脚本正常运行。若

能一并对脚本进行版本管理，当修改后的脚本发生错误时，能迅速地退回到上一个能正确运行的版本（图6）。如果一个人管理所有脚本，可能无法体会到这样做的意义。但是，工作繁忙时或者由多人管理脚本时，就体现出其重要性了。

●图6　把本地确认动作正常的文件交给Jenkins运行



特别是由多人管理时，难免会出现由于同时编辑JOB而造成的编写、保存冲突等问题，浪费大量的时间。如果使用版本管理，就能和已经提交的文件进行比较，也不会因为别人的修改造成的运行失败而紧张了。

管理中出现问题是难以避免的，关键在于出现问题后能否迅速地进行复原处理。最好在本地保留脚本文件的备份。

用批处理文件记载JOB处理?

在记载JOB处理的输入框中使用批处理文件和shell脚本，哪个比较好用呢？没有具体的标准规定。如果只是反复调用命令，用哪个都一样，所以选择自己比较熟悉的就行。

笔者常用shell脚本。虽然也能够在Windows命令提示符下通过设置环境变量进行文字处理，或者用if等语句控制处理流程，但是在Cygwin环境下同样能使用UNIX命令，加上shell脚本

Column 2
Windows环境下的话题
——命令行的开头多个文字消失的奇怪现象

笔者常用的是Cygwin环境，但有时也需要执行Windows的批处理命令。因为在Windows平台上，使用Windows命令行是难以避免的。笔者之所以如此反感，是因为使用批处理命令时，在成批处理脚本中加入汉字会造成JOB运行失败。并且批处理文件中的换行符不是Windows标准的CRLF而是LF，所以二者重复使用就无法顺利运行。

可能你会想在执行标签中添加表示当前运行状态的信息，比如在操作的各个阶段，用echo输出“调用视频转换工具”、“提交文件”等提示信息，但这是不可取的。添加这样的信息后，虽然能够正常保存，但是在运行时，会出现错误报告“刚才输入的内容不是内部命令或者外部命令，不是可操作程序或者批处理文件”，JOB运行失败。

为了不在这样的事情上浪费时间，请不要在网页上直接记载批处理命令。改用批处理文件就能解决上述问题。

中if语句比较好用，所以使用批处理文件的机会比较少。而且通过插件还能够用Python或者Ruby来记载JOB，这样的开发环境也很便捷。

③不能查看JOB的处理结果

接着，笔者考虑的是如何让其他人也能够查看JOB的处理结果。

从控制面板中可以看到Jenkins的JOB列表。上面显示有JOB的名称及邻近JOB的运行结果。从左侧的“任务状态”可以看到所有运行中的JOB，点击JOB名称下方的图标就能确认JOB运行过程中的控制台输出。

如果出现错误，其实可以通过控制台输出的错误内容来确认问题所在，但不熟悉Jenkins的人却不知道如何处理控制台输出的错误信息。结果，程序员以外的人还是无法自行确认JOB的结果，每次出现问题只能向程序员求助……这是笔者实际运用后的心得体会。

分标签记载JOB，尽量做到简洁易懂

笔者思考了上述问题的原因，决定先修改Jenkins的JOB列表。默认情况下，所有的JOB都在all标签下。

如果只有一个标签，增加JOB会纵向延长列表，找到对应的JOB相当费时。所以，当JOB较多时，可以通过建立all以外的标签将同类JOB归入同一标签下。这样，比起只有一个标签看起来要方便得多。建立数据相关的JOB标签等多个标签，开发人员根据自己的工作查看对应标签，也许就能很快地找到想要的JOB了。随着项目中JOB数量的增加，需要花费精力新建和管理JOB的对应标签（图7）。

启动JOB

笔者是从JOB列表画面来启动JOB的。如果使用这样的启动方式有困难的话，就不得不考虑其他的方法。

有一个最简单的方法，Jenkins能从外部启动JOB，所以可以在外部页面中添加启动JOB用的链接。但冷静思考一下，这个办法和从JOB列表启动没有实质性的区别。

于是笔者放弃了通过浏览器来启动JOB的想法。又考虑着是不是可以像大家一样尝试着通过Skype来启动。根据Jenkins指南中记载的外部启动JOB的方法，在Skype中建立控制Jenkins的聊天组，接受并执行该聊天组发出的命令。如果JOB的开始/结束、错误报告等信息也反馈到聊天组中，这样只需要确认聊天组的信息就能把握JOB的状态了。

●图7　汇总各个标签显示的JOB



从外部启动JOB

Jenkins提供了Remote access API[1]。如前所述，使用这组API就能从外部来启动JOB。最终，笔者想使用这组API，通过该公司的设计师们习惯了的Skypebot工具来启动JOB，通知执行的结果。

上述所设想的JOB启动方法，请参见API的说明页面。如果在JOB画面的URL后有/build字样，该JOB就为启动模式。具体格式为JENKINS_URL/job/JOBNAME/build。无论在哪里，只要在浏览器的JOB画面的URL后输入/build，就能简单地启动JOB了。然而，前提是没有使用Jenkins用户认证。

用户认证的设置

在实际使用过程中，可能会发生各种意想不到的情况。Jenkins的用户较多，有时还需要允许外部访问Jenkins。在这种情况下，尽可能选中“Jenkins管理”–“Configure Global Security”中的“启用安全”，在“访问控制”中设置用户认证（图8）。

此项设置不为尚未登录的用户提供Jenkins和JOB的链接，因此能防止JOB和数据被错误删除的情况。在向外网公开Jenkins的情况下，必须对访问的外部IP地址进行限制，这时可以和网络管理者进行沟通。

在设置了用户认证的Jenkins环境下，需要在外部启动JOB的URL中添加访问令牌（图9）。

设置为令牌的字符串没有什么特殊的限制。虽然令牌的字符串会以明文的形式记录在外部调用的脚本中，但是作为访问Jenkins的必要口令，应该避免像aaa这样简单的字符串。

用浏览器以外的方法来启动

因为可以通过浏览器来启动JOB，所以笔者又做了一个实验，尝试使用Cygwin的wget命令来启动JOB。因为是在用户认证的环境下，所以在JOB中附加了访问令牌。

●图8　设置Configure Global Security



●图9　设定JOB的访问令牌



① https://wiki.jenkins-ci.org/display/JENKINS/Remote+access+API

```
$ wget http://localhost:8080/job/DATA_
Compile/build?token=compile  -o NUL
```

因为在本地Jenkins环境中启动JOB，所以使用了localhost。而在实际环境中启动JOB，则需要修改URL以及端口编号。

Wget是以取得互联网上文件为目的的UNIX命令。原本用于指定URL和协议、下载文件，而这次只指定URL并进行访问，不保存下载文件。再通过Jenkins的界面确认，如果JOB已经运行，那么实验就成功了。实际使用的时候，如果在NUL之后添加--quiet选项，这样就不会显示wget的消息了。

在Skypebot中使用

既然可以这样启动JOB，那么就可以结合第1回中介绍的Skypebot一起使用。Skypebot常驻于Skype中，接收并执行命令。接受命令部分和执行处理部分都已经实现了，只需要在处理中使用wget，追加访问Jenkins即可。最终，通过Skype发送例如@cnv这样由@开头的字符串，就能进行处理了。具体参见图10。

关于Ruby脚本

Skypebot本身由Ruby生成。为了用Ruby读写Skype的内容，使用了RubySkype[②]这样一款插件。该插件可以通过Gem来安装。

```
$ gem install Ruby4Skype
```

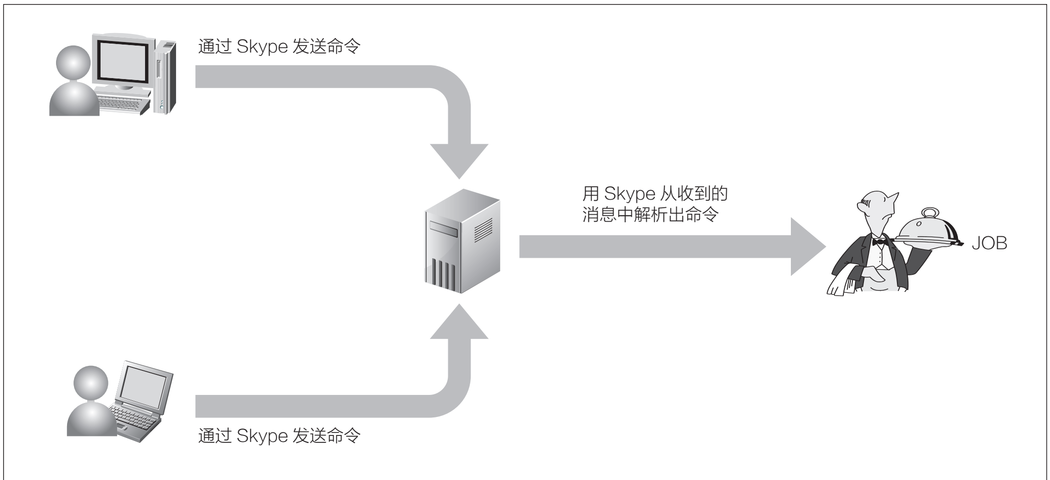
还有比这款更新的插件，但是一直没有找到升级版，所以只能先这么用着。

虽然需要这款swin.so插件[③]，但是该插件使用了Ruby1.8的库，所以Jenkins上的Ruby也必须相应地更新到1.8版本。

让我们看一下实际的脚本（代码清单1）。脚本先从Skype接收命令，再用wget启动JOB。

第7行到第9行进行插件的初始化，第11行到第18行获取主题名称为Jenkins的聊天板。这

●图10　Jenkins结合Skype运行示意图



② https://rubygems.org/gems/Ruby4Skype
③ http://www.osk.3web.ne.jp/~nyasu/vruby/vrproject-e.html

样，在其他聊天板中输入字符串时也不会发生任何反应。

从第20行开始，在数组中设置命令以及与之相应的JOB对象。从第25行开始，从skype接收命令，并且自身成为调用Jenkins的JOB。

● 运行脚本

接下来，只需要启动运行有Jenkins的电脑，启动并驻留上述脚本，就能用Skype来启动JOB了。启动脚本时，如果在聊天板上输入命令，就能改善JOB的运行。这样，如果您想了解JOB列表，就可以直接查看了。

但是仅仅这样的话，还是无法获知JOB的运行结果。在代码清单1脚本的基础上稍加修改，通过TCP端口向skype发送消息。具体而言，将TCP端口从6001修改为6601，第20行之后按照代码清单2进行修改。

驻留修改后的脚本，在TCP6601端口中书写的消息就能在skype上显示了。

至此，为设计师们构建的环境就算是完成了。通过Skype向Jenkins发送例如@cnv这样的命令来启动JOB。JOB处理开始时，向skype发送JOB名称、编译序号、确认控制台输出用的

●代码清单1　从Skype接收命令，用wget启动JOB的脚本

```
001 : #!/usr/bin/ruby -Ku
002 : require 'rubygems'
003 : require 'Skype'
004 : require 'kconv'
005 : require "socket"
006 :
007 : Skype.init 'botbot'
008 : Skype.start_messageloop
009 : Skype.attach_wait
010 :
011 : chats = Skype.searchRecentChats
012 : chats.each do |n|
013 :   topic   = n.get_topic
014 :   members = n.get_members
015 :   if topic =~ /Jenkins/
016 :     $tcp = TCPServer.open(6001)
017 :   end
018 : end
019 :
020 : JOBS={
021 :   'cnv'   => 'DATA_Compile',
022 :   'color' => 'DATA_ReduceColor',
023 : }
024 :
025 : Skype::ChatMessage.set_notify do |chatmessage, property, value|
026 :   if (value == 'RECEIVED' || value == 'SENT') && property == :status
027 :     if /^¥@/ =~ bd
028 :       bd = bd.slice!(1,bd.size)
029 :       bd.downcase!
030 :       if JOBS.key?(bd)
031 :         chatmessage.get_chat.send_message "开始#{bd}"
032 :         kick_job="#{WGET} #{JENKINS_HOST}/job/#{jobname}/build?"
033 :         system(kick_job)
034 :       end
035 :     end
036 :   end
037 : end
```

URL；处理结束后，再次向skype发送JOB名称。这样就可以知道JOB的运行状态，如果点击确认控制台输出用的URL，还可以确认控制台输出的运行日志。

笔者稍微修改了一下视频转换器的脚本，当出现错误的时候，能够向skype发送错误内容。打开6601端口，发送错误消息和原因。虽然这只是最低限度的修改，但通过这些做法，可以使运行JOB的人以及同一聊天组内的成员收到消息，一起分析错误原因。大家共同分担烦恼，真好！

终于可以在不操作Jenkins界面的情况下监视JOB的运行状态了。虽然花了一些功夫，但总算帮大家解决了“看JOB界面好麻烦”的烦恼。

●代码清单2　代码清单1的修改代码

```
020 : while true
021 :   Thread.start($tcp.accept) do |s|
022 :     while s.gets
023 :       $chat.send_message($_)
024 :     end
025 :     s.close
026 :   end
027 : end
```

本回总结

本回就Jenkins工作区的构建、JOB相关的话题以及Jenkins和Skype的联动环境集成这三方面进行了说明。

笔者并非极力推荐共享工作区，但是在必须节约磁盘空间的情况下，共享工作区还是不错的选择。本回中，因为不希望在多个地方保存大数据文件，所以将工作区设置为共享。除此之外，在开发过程中，如果遇到难以增加磁盘容量或者并非正式测试阶段的环境，共享工作区是非常有效的。构建共享工作区可能无法解决所有的问题，但是作为解决方案之一，可供我们参考。

确实从简洁的构建和管理角度来看，JOB处理都具有优势。但是在实际使用中，拼接起来的JOB处理内容显得琐碎、复杂。这时，比起网页上的输入框，使用熟悉的文本编辑器来编辑JOB的处理不是更方便吗？不小心删掉了一行处理并且进行了保存，等到发现的时候已经无法复原或者需要长时间复原。遇到这样的情况会很麻烦，所以要采取对策来预防这些事情的发生（笔者过去亲历过这种事情）。

很多情况下，JOB结果的确认会对操作产生影响。因此，希望操作员们一起确认并找出错误的原因。但是，令人遗憾的是有些人不会这么做。他们要么轻视处理的过程，要么认为数据没有做好是程序的问题。碰到这样的事可就要费一番周折了。

最初，在Jenkins中进行设置，当JOB失败的时候会发送邮件通知。但当邮件积攒得越来越多时，有些人就开始不看邮件了。就算看了邮件，也因为邮件中只记载错误而无法通过邮件探究错误的原因，就对问题置之不理。为了避免这种事情的发生，需要通过聊天工具发送日志及错误报告。出现错误时，聊天组内的成员都能收到错误报告，可以帮助当事人一同解决问题。其带来的效果，正是笔者所期望的。

其他的解决方案中也使用到了Jenkins插件，下回将以插件为话题，进行重点探讨。

分布式数据库“未来工房”

草庵辞旧主
从此居中多倩影

第2回 何为高可用性
——Riak功能探究与测试

文/上西康太（Basho Japan 股份有限公司　kota@basho.com）
译/苏祎

上一回，我们介绍了Riak的另类使用方法。但其原始设计思想、API、使用方法等介绍都比较有限，很多读者可能还没很好地了解Riak。起初，Riak是Basho的工程师为了在自己的服务上使用而创造出来的，它的基本特点为：

- 扩展性——添加服务器后，容量、性能都能相应地增强。
- 可用性——压力超过负荷时也能稳定地运行、响应。
- 容错性——发生多个故障时不丢失数据、不停止服务。

由此可见，维持系统稳定是Riak最主要的目标。 高可用性，是Riak的基础特性，本文将以此为着眼点，介绍Riak是基于怎样的目标设计出来的。

Riak存活监控的机制

首先我们来介绍Riak的基础构成。分布式数据库的构成基本上都由“选择并判断数据服务器”和“存放数据到选择好的服务器中”这两个步骤来完成。首先，先介绍选择并判断数据服务器的方法。

Riak通过bucket名和key来确定应该保存数据的服务器。ring size是一个常数。ring是分区（partition）序号和服务器地址的对照表。分区序号指的是把2^{160}的整数空间用ring size分割为常数个，这些区间按顺序被分配到的序号。（注：这个常数被称为ring size）。各个分割开的空间被称为分区。实际存放数据的实体被称为vnode。存放在Riak中的数据基本上以vnode为单位进行管理。实际应该存放的服务器，根据bucket名b和key k，计算160位的整数的hash函数Hash(b, k)，ringsize假定为N_R的情况下，按公式

$$\frac{2^{160}n}{N_R} < \mathrm{Hash}(b, k) < \frac{2^{160}(n+1)}{N_R}$$

b……桶
k……键
N_R……ringsize
n……分区ID
Hash(b, k)……hash函数

首先计算n。Riak用n、n+1、n+2来表示应复制数据的vnode。然后，以该分区序号为基础，查找ring，获得应该存储数据的服务器的地址。这个就相当于从ring顺时针挑选出3个分区。

实际上在发送请求时，还要考虑该服务器是否还在运行。Riak经常对组成集群的服务器进行存活监控。如果遇到ring中的服务器突然不能访问的情况，就从ring上顺时针选出一台应该访问的服务器（图1）。

Erlang/OTP的存活监控

应该如何实现集群中服务器互相监控呢？并不存在其他专门负责监控的分布式数据库服务器，基本上是用N对N的形式互相监控。这其实不是Riak的功能，而是使用了Erlang/OTP的功能（distributed Erlang，分布式Erlang）。

分布式Erlang[①]默认每15秒交换KEEPALIVE通信，若最后一次超过60秒，这个节点就被认为退出了集群。接下来，我们来确认一下实际中分布式Erlang存活监控的行为。一台服务器上如果针对不同进程、不同操作系统，也可以通过内核检查出linkdown等，例如就算只有某个进程终止，也能很快被检查出来。

为了再现实际发生的silent故障，需要验证物理服务器网线的连接状况。实验方案就是，以图2这样的结构，在左右的服务器使用分布式Erlang连接的情况下，拔掉*部分的网线看看会发生什么情况。

首先，在A服务器上启动Erlang（画面1）。使用-name指定集群上的标识符和IP地址。-setcookie是Erlang进程互相确认连接可行性的信息，若不一致，就不能建立连接。在B服务器上已经预先启动好了Erlang。

接着，在A服务器上向B服务器发送分布式Erlang的ping。

画面2的net_adm:ping/1虽然是名为ping，但包含了若未与B建立连接，就与其建立连接的动作。如果能正常和B服务器连接的话，会返回pong。如果不能连接的话，会立即返回pang（注：这里使用ping来检查是否已经成为集群的一员，并不是用来向对方发送包来确认连接情况。关于集群成员会在后面说明）。

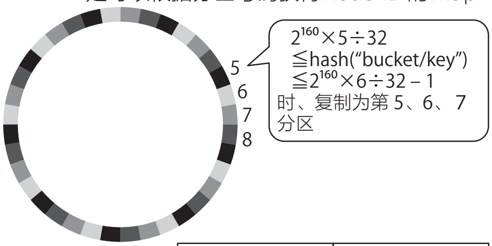
分布式Erlang非常优秀，RPC（Remote Procedure Call，远程过程调用）的调用形式也和普通函数调用相同。例如，在A服务器执行画面3的代码，在B服务器的控制台上会如下显示。

```
(dev@192.168.100.128)1> ["hogehoge~n"]
```

这样，通过使用分布式Erlang，可以方便

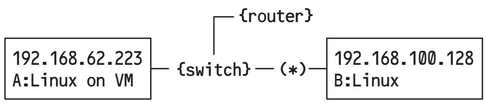
① http://www.erlang.org/doc/reference_manual/distributed.html

▼图1　找到key位置的机制。被称为ring的数据就是可以根据分区号码获得node ID的map



分区号码	节点名
…	…
5	riak@10.0.0.31
6	riak@10.0.0.27
7	riak@10.0.0.25
…	

▼图2



▼画面1　在A服务器上启动Erlang

```
$ erl -name dev@192.168.100.128 -setcookie foo
Erlang R15B03 (erts-5.9.3.1) [source] [64-bit] [smp:8:8] [async-threads:0] [kernel- ?
poll:false]

Eshell V5.9.1  (abort with ^G)
(dev@192.168.100.128)1>
```

地在远程机器上启动程序。并且，能向远程机器清晰地发送消息这样的优秀功能也有很多，Riak在多处使用了分布式Erlang的功能。

查看net_kernel模块的manpage，就可以看出它的大致设计。其中最重要的是使用Ticktime概念，它主要是检查出延时（timeout）的计时器（Timer）。这个间隔可以通过net_kernel:set_net_ticktime/2来设置。

在实际中，Erlang/OTP以15秒为间隔交换Keepalive消息。接着，在最后一次成功keepalive的60秒后，若没有任何响应，就认为这个节点的通信中断了。

我们来实际验证一下这个动作。首先使用erlang:monitor_node/2，让A服务器监控B服务器，接着，观察拔掉网线后的情况。我们的预想是，在网线拔掉大约60秒后，消息{nodedown, 'dev@192.160.100.128'}会从net_kernel发送过来，如画面4所示。

实际操作中，在时间显示21:09:30的瞬间拔掉网线，大约1分钟后，nodedown的消息过来了。这个时间有一定的误差，B服务器上也显示了如下信息。

```
=ERROR REPORT==== 19-Jun-2013::21:10:22
===
** Node 'dev@192.168.62.223' not
responding **
** Removing (timedout) connection **
```

消息中的时间，连秒数都一致纯属偶然，实际中的大规模环境里，可能不会这么精确。

如果可能的话，在相同机器上创建两个进程再验证一遍，然后与B中kill掉UNIX进程的情况进行比照。这个情况下，接收到kill的内核会立即通知A的socket，A会立即知道B已经终止。

这就是分布式Erlang的存活监控的机制。若检测到L2层以上的中断，基本应用层会立即收到通知，若是L1层级别发生了中断，大约需要1分钟才能知道。

▼ 画面2　对B发送ping

```
$ erl -name dev@192.168.62.223 -setcookie foo
Erlang R15B01 (erts-5.9.1) [source] [64-bit] [smp:2:2] [async-threads:0] [kernel-poll:false]

Eshell V5.9.1  (abort with ^G)
(dev@192.168.62.223)1> net_adm:ping('dev@192.168.100.128').
pong
```

▼ 画面3　从A这里进行RPC调用

```
(dev@192.168.62.223)2> rpc:call('dev@192.168.100.128', erlang, display, [["hogehoge~n"]]).
true
```

▼ 画面4　拔掉网线后

```
(dev@192.168.62.223)3> erlang:monitor_node('dev@192.168.100.128', true).
true
(dev@192.168.62.223)4>   erlang:display(erlang:localtime()), receive R -> {R,
erlang:localtime()} end.
{{2013,6,19},{21,9,30}}
这里拔掉192.168.100.128一侧的网线
{{nodedown,'dev@192.168.100.128'},{{2013,6,19},{21,10,22}}}
(dev@192.168.62.223)5>
=ERROR REPORT==== 19-Jun-2013::21:10:22 ===
** Node 'dev@192.168.100.128' not responding **
** Removing (timedout) connection **
```

收到通知后Riak的处理

如上描述，从分布式Erlang获知nodedown消息的Riak，调整PUT等请求的转送地址，向ring上的下一个节点执行handoff(注：handoff为将发往vnode的请求交予临时代理来处理)。使用这样的机制，除去网络上的silent故障，几乎所有的情况下，Riak都能保证访问正常运行的节点。即便是遇到silent故障，客户端等待的时间也只有60秒左右。

因为设计为没有其他特殊锁定处理，所以几乎不会让客户端的请求等待。这就是实现Riak高可用性的机制。

故障时的行为和处置

节点故障时

节点故障也分很多情况。大致可分为因HDD故障导致数据丢失的情况和其他情况。

HDD没有发生故障的情况下（即Riak用来保存数据的目录没有问题）就非常简单了。换下主板、电源、内存等故障部件，重新安装操作系统，分配一个和故障前相同的IP地址。接着，启动Riak，就和以前一样能加入到集群中了。关机的节点重新启动的话，故障期间其他服务器在Handoff中接收到的数据就开始传送过来。这种情形能通过`riak-admin transfers`来观察。

HDD发生故障，导致服务器上数据丢失的情况下，需要通过复制机（replica）来恢复数据。在这个状态下，其他服务器将其认为是临时故障，会等待相同IP地址再次启动（一般来说，难以自动区分是临时故障还是永久性故障，需要操作者在集群中指示）。替代服务器不能立即准备好的情况下，可以使用画面5这样的命令把故障的服务器强制从集群中移除。

重新设置集群内的vnode，将减少到2的复制数恢复到3。完成后，执行上期介绍的`join`命令来添加新的服务器。

如果能迅速准备好的话，最好在将故障服务器从集群中移除的同时，把新服务器加入到集群中（画面6）。

这样，vnode的重新配置处理一次就能进行了，可以节省网络上的vnode移动。

在恢复之前，PUT/GET/DELETE等基于key的检索操作仍可以正常执行，但MapReduce、listkeys等需要访问集群内所有服务器的操作，在多个节点有故障、某些vnode的所有复制都不可见的情况下，从一开始就是不能执行的。

网络故障时

交换机或路由器会偶尔发生故障，这个时候，故障交换机和路由器下的所有服务器就停止工作了。从网络的结构来看，集群中一部分的服务器由于故障变得不可见了。例如，50台服务器组成的集群中的25台由于交换机的故障变得不可见的话，剩下的能访问的25台必须和以前一样处理write和read请求。为了不让负载变为2倍，尽量调节进入的流量比较好。

▼ 画面5　集群的强制退出

```
$ riak-admin down riak@10.1.2.3
$ riak-admin cluster force-remove ?
riak@10.1.2.3
$ riak-admin cluster plan
$ riak-admin cluster commit
```

▼ 画面6　添加新的集群

```
$ riak-admin down riak@10.1.2.3
$ riak-admin cluster force-remove ?
riak@10.1.2.3
$ riak-admin cluster join riak@10.1.2.4
$ riak-admin cluster plan
$ riak-admin cluster commit
```

还有，和节点故障一样，API也会有很多变得不可用。比如，如前述例子所述若一半节点故障，一半的数据就不能读取。因为使用Consistent Hashing进行分布化，所以不能确定到底哪部分数据不能读取。还有，只要有1个不能访问的vnode，就无法获得所有的key，以及执行Mapreduce请求。

但是，数据消失或恢复时无法被清零，所以在控制流量的同时需要冷静地恢复网络。若有Riak进程停止，用Riak start来启动就可以了。

磁盘空间不足时应该如何处理

和其他很多系统一致，当磁盘剩余空间为0时，就代表是系统设计和运行的失败。为了避免这样的情况，需要慎重设计系统。

Riak也一样，在磁盘空间不足时会终止进程。

因此，当因节点磁盘空间不足而进程终止时，就会使用Handoff，调用其他节点来执行读写请求。接受Handoff的节点磁盘空间也会随之被占用，这类似于一种多米诺骨牌现象。因此，最初节点磁盘满之前，应该停止PUT和POST等请求，以免数据继续增加。若在前端有负载均衡器，可以在那里过滤掉PUT请求。

即使这样，在服务器数量较少时，vnode配置不理想或者无法及时增加节点的情况下，磁盘容量紧张的状况还是有可能发生的。这时，首先要考虑增加节点。重新配置和压缩（compaction），不需要翻倍的容量。基本上，只要保持磁盘有20%的空闲容量就可以了。

但是，删除key并不会马上空置相应的空间，处理删除请求时，Bitcask和LevelDB也只是添加数据删除用的删除标记，因此，磁盘的使用量还是会轻微增加。直到下一次进行压缩的时候，才会从磁盘上删除。所以，至少需要留有可以执行压缩的容量。

一般压缩是以heap GC（Garbage Collector，垃圾回收）算法来执行，为了执行压缩，需要有相同容量的空余容量。但是，Riak上每个DB实例以vnode为单位，并且Bitcask和LevelDB都以很小的单位进行压缩，实际上有一点空余容量就够了。若连这样空余容量都不能保证的话，最后的方法就只有删除vnode的目录了。

这种方法的前提是，为了在节点重新配置的过程中，能稳定启动Riak的进程，需要把复制节点从3个减少为2个来空出空间，在进程启动过程中，再添加节点空出空间。此方法风险非常高，是万不得已时的最后选择。

什么是可用性

前面，我们已经描述了Riak故障时采取的行为、存活监控的机制和处理方法。这些都是现实的问题，是开发、运行系统时必须考虑的问题。和其他的分布式数据库比较起来，Riak没有主从的功能分割，因此故障时处理的模式也比较少。

CAP定理和Riak可用性

CAP定理说明了**一致性、可用性、容错性**这三个特性不可能同时实现。虽然也有专门证明它的论文，但一般都认为它是分布式系统方面的经验法则。Riak在这之中关注于可用性，以不论何时都能写入、读出作为其根本设计思想。也不能说其他两个性质完全没有，但一定程度上那两性质取决于用户如何实现。

一致性是一个旧瓶装新酒的问题，基本上为了保证多个复制的一致性、加锁、2段提交等需要多个请求在多个状态等待的处理。因此会出现等待时间，不能保证稳定延迟的情况也会

出现。还有，为了保证容错性，就必须保证系统全体是未被分开的（例如，不存在多个主服务器，同一个数据不允许在多个地方进行更新），必须是类似于HBase和BigTable的中央管理型系统结构。

相反，对这些特性稍稍妥协的话，设计分布式系统就能非常简单了。不使用锁，没有复杂的等待也就几乎没有被block的处理，延迟也非常稳定。还有，若不准备主服务器，也不会有主服务器的故障、恢复引起的全体服务器的停止，也不需要设计metadata的事务。针对这些问题，去掉不需要的复杂设计，追求可用性的结果，就是现在Riak的设计。

总结——为了实现高度可用系统

至此，本回介绍了Riak存活监控的机制、设计的考虑要点和各个故障的对应方针。

最重要的是事先的准备工作。最初的集群一定是预算很少的，使用5台小服务器来搭建也是可以的。在大负荷和数据增加之前，大幅提高系统的容量是非常重要的。针对这点的最佳实践[②]也准备好了官方文档[③]。

使用Riak的话，可以不停止集群，执行各种升级工作。服务器就不用说了，做得好的话，网络的更新也可以只停止部分系统来完成吧。

② http://docs.basho.com/riak/latest/cookbooks/best-practices/
③ 参考URL
http://www.cse.psu.edu/~gcao/teach/513-00/c7.pdf
http://www.cs.duke.edu/courses/fall07/cps212/consensus.pdf
http://rodin.cs.ncl.ac.uk/Publications/avizienis.pdf
http://www.erlang.org/doc/man/net_kernel.html#set_net_ticktime-2
http://docs.basho.com/riak/latest/cookbooks/Linux-Performance-Tuning/

安全实践鬼手诀

铃木弘信
suzuki.hironobu@gmail.com
译/honnkyou

【第二回】当今流行的目标式攻击和APT攻击

这一回，让我们聊聊目标式攻击和APT（Advanced Persistent Threat）攻击，它们被称为新型攻击。

计算机入侵历史

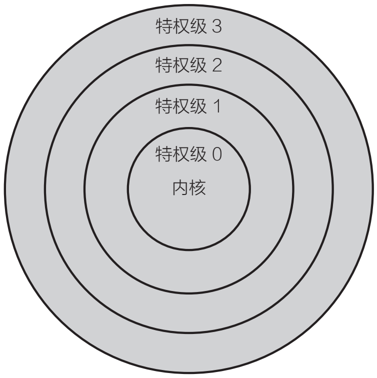
对计算机的入侵以及反入侵对策是如何发展到今天这个程度的呢？让我们先来回顾一下历史。

Sneakers

在20世纪50年代至70年代的计算机诞生初期，计算机是很昂贵的器材，只有操作员和极少数特权用户才能接触到并操作这个特殊仪器。因此，初期的计算机内都存放着跟其价值相称的重要信息，有军事系统、银行账户管理系统、政府机关处理的国家级数据。要处理重要信息，必然要有信息保护功能（图1）。这个时期的计算机被称为通用机，比现在的PC机要早出现很长时间。

在那个时代，所谓的入侵计算机是有特定的意图、对特定目标进行的行为。当然，限制

◆图1　特权级保护机制（hierarchical protection domains）



使用权限分层，每层只能赋予最小权限，这个想法在 1964 年就被 Multics 采用了。
特权级保护机制的核心思想就是分层地持有对计算机资源的访问权限。图中特权级 0 是可以访问任何资源的最大权限，越向外环权限越小。针对不同的使用级别赋予其适当的权限（Privilege），特权级 0 对应的是内核，特权级 1 与特权级 2 对应的是设备驱动，特权级 3 对应的是应用软件。
例如，UNIX 的对应关系就很典型，特权级 0 对应内核模式，环 3 对应用户模式。

访问终端等物理级别的限制也是有的。

推荐大家欣赏一下名为Sneakers（1992）的电影[①]，这部电影中出现了很多盗取情报的技术。进入90年代，该题材很多作品都是粗制滥造，比起那些把普通cracker故意称为hacker的山寨电影，这部电影相当不错。

20世纪80年代中期，今天的PC机出现了。不久，PC机就普及开来。

这时，重要的信息不再放在通用机中，而是逐渐分散到了PC机中，人们称这种变化为小型化。但是，关于计算机安全，PC机并不是小型的通用机，刚好相反，它没有保留通用机具备的保护重要信息的功能，而是变成了接上电源后连密码都不需要的、任何人都可以随时使用的、“方便”的工具。

PC机最开始设定为只能一个人使用，没有配备非常严格的访问权限限制，里面的软件可以访问任何系统资源。另外，对PC机资源的使用没有任何限制，可以进行任何作业，这就是PC机的“方便”之处。

过了不久，PC机发展到可以连接到本地局域网的程度。这个时候，如果通过网络任何人都可以随便访问PC机就容易带来麻烦，因此强化了访问权限控制。但是根本问题并没有解决，没有统一的访问控制策略，PC机还是一个很脆弱的系统。不过，PC机底层软件经过不知几代的发展，终于还是形成了统一的访问控制安全策略。

但是，这一策略在初期也没有得到很好的贯彻。好不容易在技术层面上实现了将管理者权限和一般用户权限分离的想法，可是大部分用户无论做什么仍是在管理者权限下进行。系统即使在功能层面满足了安全条件，想要改变用户已成为习惯的不规范操作行为也是十分困难的。

近年来PC机底层软件的安全性得到了相当大的提升，但是PC机上搭载的第三方软件也越来越多。搭载的第三方软件与底层软件相比安全性等各方面都不高，这成为了入侵计算机的突破口，由第三方漏洞入侵系统的事情常常发生。

另外，还有一些组织由于诸多原因，现在还不得不使用从前的不安全底层软件，从而面临诸多隐患。

彩虹安全体系与SELinux

最近新闻纷纷报道NSA（美国国家安全局）收集SNS记录和电话记录来监视国民的棱镜计划。不过对于NSA来说，出于保障国家安全的要求，有责任保护计算机安全。

美国国防部的计算机安全中心和NSA的国家计算机安全中心等机构在1983年到1993年间，以文件的形式规定了互联网以及计算机应该具备的安全标准。根据文件的颜色，又被称为橙皮书（DoDTrusted Computer System Evaluation Criteria，5200.28-STD）和红皮书（Trusted Network Interpretation，NCSCTG-005）[②]。

以这些文件为基础，National Security Agency’s Trusted UNIX（TRUSIX）Working Group制定了访问控制的标准，而实现了该标准的系统就是SELinux。该系统引入了在处理军事机密和政府机密时需要具备并达到了安全级别的访问控制标准。如果系统没有搭载这个标准，就不可能

① Youtube上的标题为“Sneakers（1/9）Movie CLIP-Professional Bank Robber（1992）HD”。由罗伯特·雷德福/西德尼·波蒂埃/丹·艾克罗伊德/瑞凡 · 菲尼克斯/本 · 金斯利等主演。

② 书的颜色种类众多，所以又被合称为彩虹丛书（Rainbow Book）

导入到军队和掌握重要情报的政府机关中。

这样看似在一定程度上解决了问题，但实际却并非如此。SELinux的设置只要稍有冲突就无法运行，所以那些介绍开源服务器应用程序安装方法的博客，一般都会建议先将SELinux功能设定为无效再开始后续动作。

要想在SELinux这类系统中开发能够任意运行的恶意软件是非常困难的。但安全是安全了，相应的代价就是安装正常的软件也相当费力。在很多独立的库或框架相互关联的情况下，对必要的文件设置合适的安全上下文并毫无冲突地顺利启动是相当困难的。

这是因为各种应用程序和系统不是以SELinux，而是以经典的Unix访问模块（Discretionary Access Control）为基础开发出来的。作为权宜之计在SELinux中可以适用，可一旦设置有一点错误或者遗漏，系统就运行不了了。笔者在博客中分享过在运行了SELinux的CentOS 6中安装Redmine 1.3的经验③，安装确实要花费更多的时间。

从这些经验来看，首先将SELinux功能关闭再安装也是没有办法的事情。因此，希望SELinux及另一个Linux系统安全应用程序AppArmor（应用程序访问控制系统）能被广泛使用。

但同时，安全到底还是要在成本与便捷性之间找到平衡，而在什么地方找到平衡却是个很难的问题。现实情况下无法一概而论，这正是解决安全问题的难点。

客户端PC机成为服务器的威胁

即使服务器端设定得犹如铜墙铁壁般坚固，但如果登入该服务器的客户端PC机安全管理不到位的话，服务器端也相当于没有防备。利用这一点进行攻击的是2010年出现的Gumblar，这种攻击行为的主要目标是服务器，攻击PC客户端只不过是前奏而已。

首先被感染的是PC客户端，Gumblar会探查使用明文保存了客户端所连接的服务器名及密码的应用程序注册表和设定文档，随后窃取情报。

Gumblar抓住了第三方应用程序欠缺安全性考虑且信息管理混乱的盲点。2013年，像BHEK2这种以Java、Adobe Flash、Adobe Reader等第三方应用程序为目标的攻击开始出现④。

只接受SSH公开密钥连接等，服务器端能采用的应对方式十分有限。说到底，如果客户端有漏洞的话，服务器端最终都会受牵连而被攻击吧。

中学生的网络战争

2013年春季，朝鲜的对外宣传网站被“匿名者”黑客组织攻破，大量信息被窃取导致会员名册等外流，相信大家对媒体将其称为网络战争并大肆报道的事还记忆犹新吧。但是，答案揭晓后却发现这居然是韩国一名中学生所为。

或许你会想一名中学生怎么可能办到，但其实原因很简单。单纯入侵服务器的话，无论

③ CentOS下安装Redmine 1.3　http://h2np.net/docs/redmine-selinux.html

④ IIJ-SECT Security Diary博客“日本国内使用BHEK2恶意篡改事件报道”中有详细介绍，请参考https://sect.iij.ad.jp/d/2013/03/225209.html。

是工具还是教程，网络上都广泛流行。即使没有计算机专业知识，只要参照着某个博客中的详尽说明一步一步执行，就能入侵别人的服务器。

另外，要是受攻击的服务器这一方恰巧找了一位既没相关知识又没经验的人来管理服务器，现学现卖地安装软件的话，那就成了再好不过的靶子了吧。

将这种程度的行为比做军事事件，称其为网络战争，特意夸大其危险性，这么做只能说是没有认清问题的本质，即不管怎样，现在的系统都切实存在着许多安全漏洞。而且，笔者强烈认为正是对问题本质含糊不清的认识才是隐患所在。

无序的组织与混乱的攻击

在很早之前，被自我表现欲控制、想在网络中显露自己的人就大有人在，这样的人将来也会一直存在吧。另一方面，从前与现在最大的不同点是可以倒卖窃取的信息，这也是牟利手段在不断“进步”的一种表现。换种说法说的话就是以入侵系统为职业的人越来越多。

过去可以赚到钱的，如spam从业者、盗取信用卡信息、擅自盗用国际电话或付费电话等，都是一些很直接的、没有什么附加价值的行为。

现在的现实则是，(出于政治目的)“匿名者”黑客组织发布朝鲜网站上的信息，维基解密窃取组织内部的信息并公诸于众等。可以肯定的是，用相同的手段盗取信息并将其变卖的组织也一定存在。如果有信息买家存在，信用卡号、spam名单等也会成为带有高附加值的商品吧。

出于军事动机窃取情报的行为也并不是多么难以想象，可以说这也是一种对附加价值商品的窃取。不单单如此，2011年德国的独立司法部门甚至允许并实行了以搜查为目的制作并使用恶意软件的规定⑤。

笔者认为，以目前的情况来看，潜在的攻击参与者非常多，究竟是谁在做什么完全看不清，也就是说目前是一个相当混乱的环境。作为保护方如果预测不到攻击由哪个方向过来是一件相当麻烦的事情。

Advanced Persistent Threat

APT直译过来就是“高级持续性威胁”，NIST（美国国家标准与技术研究院）文件⑥中说明了APT的特征，总结来说有以下几点。

NIST定义的APT特征：

- 攻击者具有十分强的技术能力及资源。
- 攻击者不只在网络上进行攻击，还有物理攻击或者骗术(欺诈/欺瞒)等多种攻击手段。
- 能够入侵目标组织内部，确保实施攻击行动的立足点。
- 能够实施从内部窃取重要信息或以控制目标为目的行为。
- 能够潜伏，直至达成目的。

具有这些特点的行为，明显与刚才中学生的行为及其目的级别不同。但是，在APT说明中加入了几点要素，结论也变得不容易明白了。比如说，究竟怎么才能在最开始受到攻击时就知道“攻击者具有十分强的技术能力及资源”呢?

另一方面，Microsoft对APT有着意味深长的论断，笔者认为该论断非常贴切。

⑤ Germany spyware：Minister calls for probe of state use http://www.bbc.co.uk/news/world-europe-15253259
⑥ Managing Information Security Risk http://csrc.nist gov/publications/nistpubs/800-39/SP800-39-final.pdf

利用综合性术语（如APT）可掩盖这一事实，并给人一种印象，即所有此类攻击在技术上极其复杂且以恶意软件为主导，从而使得更加难以规划有效的防御措施[⑦]。

凯文·米特尼克

20世纪90年代中期，一个名为凯文·米特尼克（Kevin Mitnick）的人因入侵他人计算机被逮捕。为了入侵特定的计算机，在技术方法难以实现的时候，他跟他的同伙采取了骗术，就是用欺诈的方法欺骗别人获取信息，并以这些信息为线索入侵计算机。

他们将这些行为美名其曰社会工程学，其实说穿了就是单纯的欺诈。即使是入侵技术，也并不能开发出原始的事物，使用的不过是已知的方法。但是即使如此，对于成功实施入侵来说已经足够了。最后他因败给被比他技术优秀的系统管理员而被捕。

凯文·米特尼克所做的事情就是NIST定义的APT行为。也就是说，名为APT的攻击行为并不是从近期才开始的。

目标式攻击

恶意软件入侵案例

关于目标式攻击，我们来考虑一下下面这些恶意软件入侵的案例。恶意软件入侵后的事情在这里暂不涉及，有机会我再讲解，这里先将大家领进门。

● 邮件添加实例

① 将添加了恶意软件的邮件发给指定的人。

② 因为是杀毒软件检测不到的恶意软件所以无法应对。

③ 恶意软件感染计算机内部。

● 钓鱼链接实例

① 向指定人发送含有指向恶意软件的钓鱼链接。

② 邮件内容为“您的申请出错”这类一不留神就会点击链接的内容。

③ 邮件内容被识别为文本格式，因此被杀毒软件排除在检查对象外。

④ 使用SSL连接钓鱼网站。

⑤ 钓鱼网站预先埋下的代码能够利用Adobe Flash Player、Adobe Reader、Oracle Java的漏洞感染计算机。

⑥ 恶意软件安装向导入侵到PC机内部。

⑦ 恶意软件安装向导下载真正的恶意软件。

这样，一直以来用户应该遵守的病毒对策基本原则，即“不访问可疑网站”“不浏览可疑附加文件”“经常更新杀毒软件”等原始做法的意义就不是很大了。

Google Chrome等浏览器在访问的网站中存在可疑代码时能够发出警告，但对第三方应用程序或者插件（例如PDF文件等）也并不是都能有效预警的。

那么让我们来考虑一下邮件添加和钓鱼网站的应对策略。对于前者来说，如果是杀毒软件识别不到的恶意软件那是无法应对的。考虑到恶意软件入侵，应该将阅读邮件的环境与处

⑦ Microsoft公司“目标攻击和确定的对手——来自复杂的、资源丰富的攻击者的威胁”http://www.microsoft.com/zh-cn/download/details.aspx?id=34793。英语版原标题为Determined Adversaries and Targeted Attacks。——译者注

理信息的环境明确分开。笔者就是如此，不会在自己的电脑上直接处理邮件中的附件。附件类型是应用程序特有格式的在Gmail上处理，只有在Google上确认没有问题并且想本地保存的邮件才会通过别的途径下载。

对于后者钓鱼链接的情况，因为邮件内容本身是文本格式，杀毒软件检查附件也毫无意义。即使想监视通信也会因为是SSL防御而无法实现。在恶意软件入侵后，跟外界通信时可能会检查出入侵行为，但如果没有抓住这样的机会也就只能等着被入侵了。

“不要随意点开链接”“随时更新软件及杀毒软件”等都是最低级别的防御措施，作为积极的安全对策，有必要准备在沙箱环境下的浏览器。

但是，目前为止还没有可以自动运行的沙箱环境。笔者在Emacs环境下阅读邮件，为了避免不小心点开链接，会将链接功能设定为无效。而且在对HTML邮件进行任何操作之前，都要查看一下HTML代码。

虽然计算机上“方便用户使用”的特点等于隐藏了内部信息，用户不需要具备专业知识，但这点对恶意软件来说也是最好的隐形衣。

导入SELinux或AppsArmor这类环境或许可以起到一定的作用，但也要考虑到其副作用，即大量的第三方软件可能会因为不兼容而无法运行。

将浏览器和任意应用程序自动隔离的执行环境中，使用有限的权限操作，即使存在漏洞，造成的影响也被限制在最小范围内。如果能持续不断提供这样的功能可以说既方面又安全，但这种环境的实现还需要一定时间。现在还是只能逐个手动应对，实在是一件遗憾的事情。

最　后

目标式攻击和APT攻击，或者被称为新型攻击的行为，都蕴含着攻击者强烈的动机，技术层面的东西与过去相比没有实质变化。

本回虽没有提到，不过最近到处都能听到军事意义上的网络攻击话题。但是，军事性质的网络攻击也有一个前提，那就是如前文所述，特别是以PC机为代表的计算机还不存在一种能确保其安全性的使用方法。即使一名中学生，如果将流传在非公开网站上的信息组合起来，也可以达到制作恶意软件的水平。概念新颖、真正意义上的新型恶意软件一年也出不了几个[⑧]。但另一方面，以已知漏洞为目标的恶意软件，其变种数量之大又是为什么呢？显而易见，是因为漏洞一直残存着。

基于目前脆弱的结构和管理，在极其狭窄的范围内讨论计算机防御等问题，短期来看可能有些意义，但要解决根本问题还是有相当长的路要走。

不被供应商一直提供的脆弱系统所左右，自欺欺人地使用倒退的理论和管理来回避当前的问题，而是用全新又安全的方式开发安全的系统，积极地使用进步的理论来解决这一系列问题，这才是笔者所希望的。这一过程会花费很多时间，但欲速则不达，说到底我们应该采取的还是行之有效且根本性的解决方法。

⑧ 实现了感染与攻击分离的Ramen蠕虫，以及刚刚提到的通过客户端狙击服务器的Gumar这种概念新颖的恶意软件的出现，从总体看是极其稀少的。另外，被称为“0day攻击”等利用未知漏洞进行攻击的情况也不是频繁出现。



菜鸟编程入门

开发一个iPhone阅读类应用

第4回

让图片可以根据不同机型切换

GimmiQ（GimmiQ；Itanokumanbou&lio&reverse）
http://ninebonz.net/
http://www.studioloupe.com/
插图/中川 悠京
译/苏祎

本连载希望从未编程过的人也可以体验制作APP的乐趣。此次我们将介绍让上回准备好的不同大小和分辨率的图片根据不同机型切换的方法。

往期回顾

上一回连载中，我们讨论到各种用途所需要的图片，而这一次将讲解如何编写代码使这些图片可以根据不同机型切换。一上来就深入了解每一行代码的意义并非易事，但如果能先理解“如果这样写代码，就会带来那样的结果”，就已经足够了。

随着基础知识的逐渐稳固，自然而然就会更深入地理解这些代码的意义。即便完全无法理解，觉得很难也请不要气馁，先按照本文的示例模仿写写看吧。就像不理解发动机的构造也能开车一样，程序也不需要从一开始就完全理解，先从能让它运行起来开始努力吧！

根据画面尺寸切换图片的方法

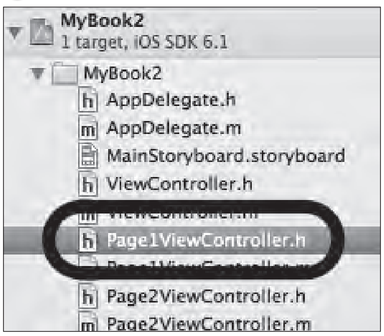
那么我们就开始正式写代码了。为了加深理解，我们可以将代码的意义比喻成“戏剧”，那程序就是“剧本”，图片和按钮则可以算是在舞台上出演的“演员”。演员必须按照剧本来进行表演，但现在（将图片拖拽到StoryBoard的阶段）还没有与演员相匹配的角色名称。就像是告诉他们“你们就站在那里什么都不用做”这样的状态。图片虽然已经有了类似Page1.png这样的固有的名称，但这个只能算是“演员名”，而不是在剧中要出演的“角色名”。现在的状态是，大家都还不知道具体要出演什么角色，所以即便读了剧本，也不知道自己应该在什么时候做什么事情。

STEP 1

从Xcode的左菜单栏里的文件夹一览中的ProjectNavigator里选择Page1ViewController.h（图step1-1）。以“.h”为扩展名结尾的文件被称作为“头文件”，比喻成戏剧的话，就是在每一场戏中记录（定义）将出演的角色的页面。在现阶段的话应该只有以下3行内容：

```
#import <UIKit/UIKit.h>
@interface Page1ViewController : UIViewController
```

step1-1



```
@end
```

不过在@interface Page1ViewController : UIViewController之后加入大括号“{”并点击 return 的话会自动空出一行并追加上结尾的大括号}。在这两个大括号之间，加入以下内容：

step1-1

```
#import <UIKit/UIKit.h>

@interface Page1ViewController : UIViewController{
    IBOutlet UIImageView *page1Image;
}
@property (nonatomic, retain) IBOutlet UIImageView *page1Image;

@end
```

```
IBOutlet UIImageView *page1Image;
```

这一行最后部分的“page1Image”代表了角色名。第一场戏（第一页）出现的就是“page1Image”这个角色了。

另外，在大括号后面继续加入：

```
@property (nonatomic, retain) IBOutlet UIImageView *page1Image;
```

变更后会变成图step1-2的样子。

STEP 2

接下来在ProjectNavigator中选择Page1ViewController.m。以“.m”结尾的文件被称为“实现文件”，是写入每个角色细节动作的部分。哪一个角色做什么事情，根据不同场景会做出怎么样的行动，都可以在此作出具体的指示。

首先按照step2-1那样，在@implementation Page1ViewController后写入：

```
@synthesize page1Image;
```

这样的话在头部文件定义的角色和实现文件就会衔接起来了。

接下来，请找到-(void)viewDidLoad的部分。这表示的是view（场面）被读取的时候，即这一场戏的开幕。在这之中加入这一场戏里会出现什么事情的说明，就可以指示在戏幕开始时希望进行的事项了。

```
- (void)viewDidLoad
{
[super viewDidLoad];
```

在这之后请输入下方的内容（参照图step2-1）

```
if (UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPhone) {
        CGRect frame = [[UIScreen mainScreen] bounds];
        if (frame.size.height==568.0) {
            [page1Image setImage:[UIImage imageNamed:@"page1-568.png"]];
        } else {
        }
}
```

if是指在“如果”的情况下怎么办，else是指在“除此之外”的情况下怎么办，这个方法可以根据状况切换不同行为。在这里，确认使用中机型的屏幕高度，若高度超过568px的话，就指示page1Image这个角色由“page1-568h.png”来替代，而非由“page1.png”来出演。

那么，可能有很多人会有疑问，为何在这里出场的不是“page1-568h@2x.png”，而是“page1-568h.png”呢？况且没有@2x的标准解析度的“page1-568h.png”在最开始(上回连载中)并没有准备，去调用在项目中不存在的文件有什么意义呢？

step2-1

```
#import "Page1ViewController.h"

@interface Page1ViewController ()

@end

@implementation Page1ViewController

@synthesize page1Image;

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    if (UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPhone) {

        CGRect frame = [[UIScreen mainScreen] bounds];
        if (frame.size.height==568.0) {
            [page1Image setImage:[UIImage imageNamed:@"page1-568h.png"]];
        } else {
        }
    }
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (IBAction)page1ReturnSegue:(UIStoryboardSegue *)segue
{
}

@end
```

之前也简单提到过，程序是以与“Pixel”按比例相对应的“Point”为基础运行的。@2x图片是仅在Retina显示屏的客户端使用时，程序会优先读取的图片。所以在这次的例子中，4英寸屏幕的客户端本身只有Retina的显示屏，所以只要指定了“page1-568h.png”，就会自动读取“page1-568h@2x.png”了。若在程序中指定的是“page1-568h@2x.png”，程序就会去寻找在这个文件名基础上再加一个@2x的“page1-568h@2x@2x.png”。但若无法找到同名文件的话，就会自动将“page1-568h@2x.png”的画面以放大2倍的状态显示，最终就会显示为原本图片的4倍大小。

读了这篇文章也还没有什么感觉的话，可以实际来尝试一下加了@2x的情况会怎样。所谓的程序，就是要通过反复尝试与出错来加深理解的。

接下来，else内并没有任何内容，是因为原本就调取了“page1.png”来充当page1Image的角色，所以在高度不是568px的时候，不需要做变更就可以进入下一步了。

进行到这里，请在第一页之后的View Controller的“.h”“.m”文件中，也将step1和step2的内容写入。写入时，“角色名”请修改为和View Controller的页数一致。比如说，Page2ViewController的话需要修改为“page2 Image”，Page3 ViewController的话就是“page3Image”，以此类推。这样，代码输入就结束了。

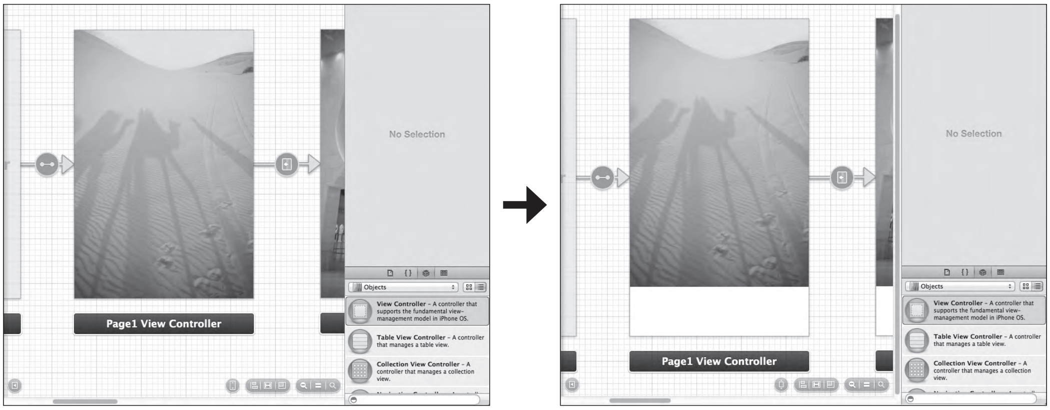
STEP 3

点击ProjectNavigator里的MainStoryboard.storyboard，在Storyboard的右下方会排列着一些透明的按钮，点击最左侧的圆形按钮。(图step3-1)。这样，应该就可以对View Controller的高度延伸状况进行确认了(图step3-2)。现在，我们便可在实际中亲眼确认，图片在4英寸屏幕的机型中展现出来的效果是如何了。

step3-1



step3-2



BEFORE　　　　AFTER

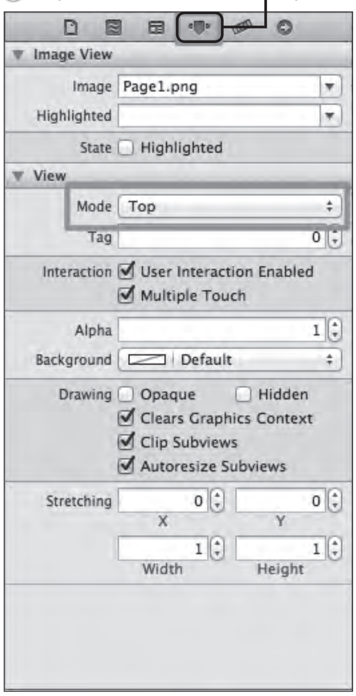
STEP 4

接下来要像图step4-1一样，在View Controller Scene中点击“Image View-Page1.png”，在Xcode右侧菜单栏的Size inspector中将Height的数值从480调整为568。

这样画面的显示范围就已经调整为4英寸了。这里若不调整为568的话，图片的内容即便是切换为4英寸的内容，但是因为显示高度只有480，所以480之上的图片内容就无法显示出来了。

但是此次画面是中间对齐的，所以在3.5英寸的时候，上面会有余白。为了修正这种情况，需要点击Attribute inspector，将Mode中现在为“Center”的部分修改为“Top”（图step4-2）。这样的话画面就会向上对齐，3.5英寸和4英寸的时候都可以正常显示了。

step4-2



step4-1



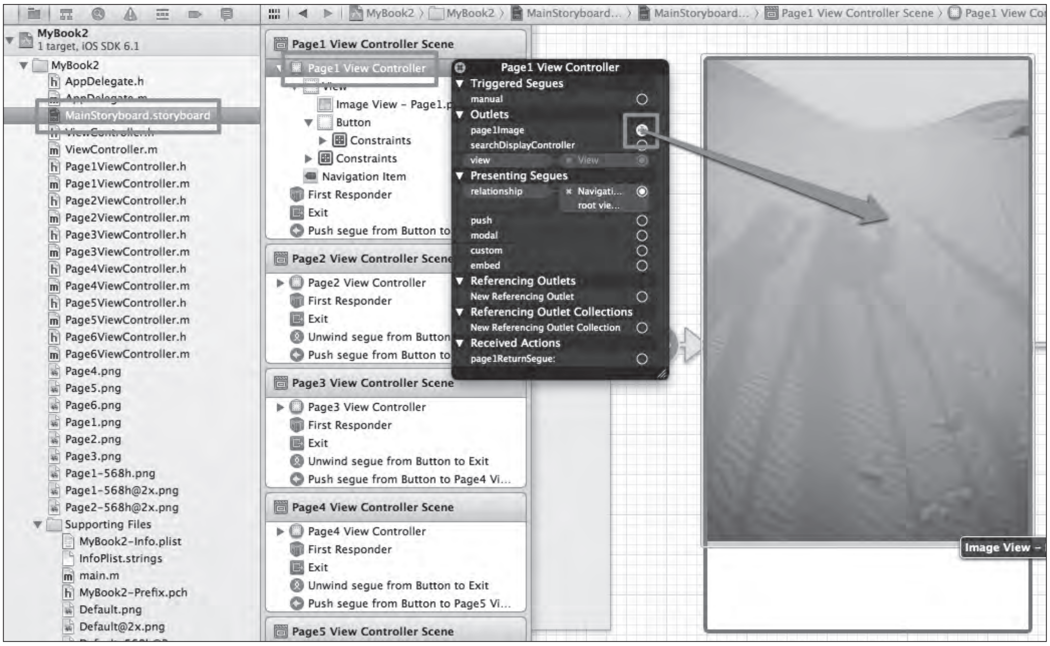
STEP 5

接下来就是角色分配的工作了。已经有了剧本了，但是还没有给每一个演员分配角色名称，所以大家还不知道自己应该在哪一场演出什么角色。现在只需要向演员告知角色名，演员们就会按照剧本来演出了。

如图step5-1，在按[control]键的同时点击Storyboard的左菜单栏的Page1 View Controller按钮，就会出现浮层菜单。在菜单的Outlets中，包含着刚才的代码所添加的“page1Image”。按住“page1Image”名字右侧的圆，直接拖拽到第一页的图片之上并放开。这样的话“page1Image”与第一页的图片就会被关联起来了。这个阶段，这张图片就被分配了角色名。

接下来就是像“Page2 View Controller”“Page3 View Controller”……这样将页数部分调整，给所有的图片分配角色名。这样的话演员和角色就相互关联起来，可以按照剧本启动程序了。

step5-1

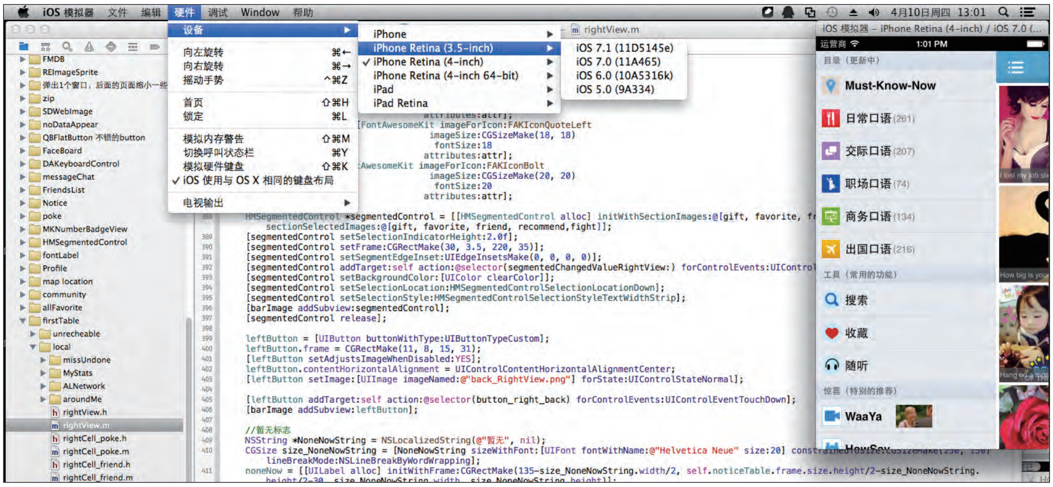


运行确认

为了测试画面是否真的会切换，点击在Xcode的窗口左上方的Run（执行）按钮，iPhone模拟器就会启动了。模拟器启动后就可以像图1一样，在菜单栏的“硬件”—“设备”处调整机型，在此处切换iPhone Retina（3.5-inch）和iPhone Retina（4-inch），确认在两种机型上会如何展示。

有关不同机型的图片切换的方法说明就结束了。iOS编程的基础，原则上就像此次这样对不同的图片、按钮等进行配置，分配角色名和任务。今后，写代码的量也会逐渐增多，所以先从基础开始好好地理解，才能更容易理解程序的构造。

▼ 图 1　模拟器中的设备切换



Column

活用注释

写程序时留下注释是非常重要的事情。注释是指在代码中自由地留下笔记。将每一部分代码的作用作为笔记留下，能便于自己或者其他人看代码时能更容易理解代码的意思。

写注释必须让程序处理，方能和代码区分开来。Objective-C 的时候，在注释的前面加入“//”，这之后的文字内容直到行尾就会被注释（Comment Out），而不被作为代码处理。

若希望让多行内容作为注释，就在希望注释的开头部分加入“/*”，结尾部分也加入“*/”，这样可以将这之间的代码全部注释。另外也有快捷方式，就是将选择范围用 command + / 整体进行注释或解除注释。

本连载的支持页面对本文进行了补充说明，请参考！
http://www.gimmiq.net/p/sd.html

Leo Rivas
Twitter @StudioLoupe

iOS 应用开发的电子绘本作家和漫画家。个人的代表作是以 StudioLoupe 的名义发布的、使用手指拖放即可保存数字的“融合计算机”（Fusion Calc）。电子绘本在 iBookstore/Kindle 商店都获得儿童书籍类综合第一名。个人主页里正在连载漫画 Hell Baseball。



Itano Kumanbow
Twitter @Kumanbow

神奈川工科大学外聘讲师。与 Leo 一起以 GimmiQ 的名义发布了 MagicReader（一款不用手就能翻页的电子书阅读应用）。个人以 NineBonz 的名义发布了 Crop It Cam！（一款小清新的照片处理应用）、“从 i 列车的车窗—对了！去京都吧！”（一款虚拟旅行应用）等。这些应用的介绍可参考应用介绍网站 http://appmaga.net/。



presented by Japan Android Group
http://www.android-group.jp/

第39回

用设计简易且高速的FPGA改变Android

Google Android作为移动设备的首个开源平台，吸引了很多工程师的目光。让我们学习那些由经验丰富的Android工程师分享的技巧和信息，朝着Android的世界大步前进吧！

文/小山 忠昭 KOYAMA Tadaaki
FPGA信息有限公司
koyama@fpga.co.jp
译/唐洪军

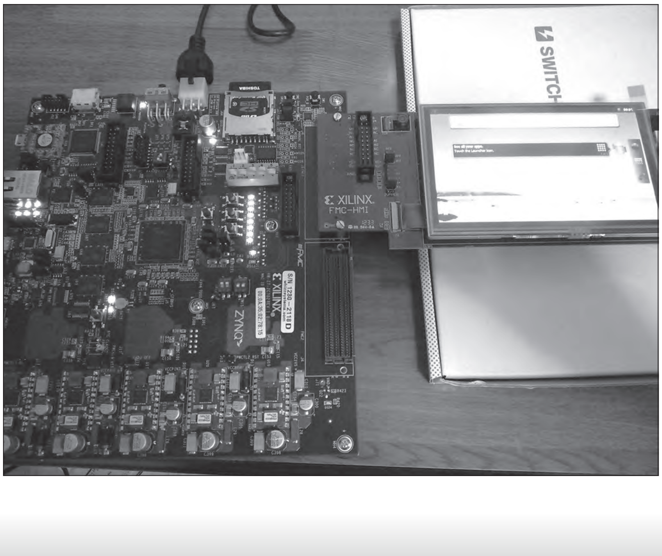
在FPGA上搭载Android

笔者试着在FPGA（后面会详细说明）上搭载了Android，请看图1。Android的版本虽然有些老，但使用的评估板是由Xilinx出品的ZC702，这款评估板上搭载了Xilinx生产的FPGA。

这款FPGA，是由FPGA的顶级制造商Xilinx公司在2011年发布、2012年上市的Zynq系列的FPGA。

以往的FPGA，因FPGA制造商和所使用CPU的独特性，一般的OS都无法运行。但Zynq系列的FPGA，不但可以运行Linux，还可以运行Android等各种各样的OS。如图1所示，在FPGA上搭载Android，得益于用户操作和I/O周边接口的发展，原因是近些年随着Android等智能手机的普及，出现了很多触屏类的用户接口设备。而且，像LAN、USB、SD卡等接口，不仅仅应用在电脑或智能手机上，在几乎所有的电子设备中都得到了广泛使用。也就是说，正因为用户接口和其他各种存储接口的完备，才得以在FPGA上搭载Android。

▼图1　搭载了FPGA的ZC702运行的Android



Zynq系列是比较新型的FPGA，搭载了Android系统之后到底可以做哪些事情，还在慢慢摸索中。现在看来，在FPGA擅长的图像处理和高速处理数据方面，它还是很值得期待的。大胆猜想一下它将来的用途，大概会是让Android系统能够搭载Kinect，或者用Android进行大数据处理等。

什么是FPGA

首先为不了解FPGA的朋友简单介绍一下。FPGA是Field Programmable Gate Array的简称，它是一种可以对集成电路进行编程，形成数字电路的装置。1985年由Xilinx发布，同年上市，随着性能的不断提升，一直发展到现在。

一般说起程序，都会想到它是由CPU来执行的。经JAVA和C语言这种高级语言编译后，会生成机器语言，然后再由CPU来逐行执行。与之相对的，FPGA使用的是一种叫做硬件描述语言的面向硬件的程序语言，编译之后，转换成FPGA的内部数据，这些内部数据用来表示数字电路本身。

FPGA接电之后，读取自身的内部数据，构成数字电路。按照这些内部数据进行数据处理，因此，必要的处理可以同时进行，执行速度是CPU的100倍。经过优化之后，可以达到CPU的10 000倍。

FPGA实际使用的领域也非常广泛，对于现在电子电路技术，FPGA已经是不可缺少的重要部分。FPGA产品的特性，趋向于少量、多品种化。它主要面向产业，也在一部分民生产品上使用。因为在通信和图像处理方面比较有优势，手机的基站和制造工厂的检查装置也经常使用。除此之外，一些档次较高的电视、音响、电车、人造卫星里，都有FPGA的身影。甚至游戏机中使用的KINECT里也有使用。另外，比较特别的是，在金融领域里也会应用FPGA，因为只有CPU的话，很多时候都难以跟上快速变化的市场数据，所以他们使用FPGA进行快速的演算，然后再使用FPGA的演算结果。现在FPGA似乎也有朝着在汽车制造业进军的趋势，开始应用于不会相撞的汽车等，搭载了图像处理功能的车辆中。

FPGA设计的现状

FPGA的处理速度很快，但知名度却很低。人们并不常使用FPAG，笔者觉得原因主要有以下几个方面。

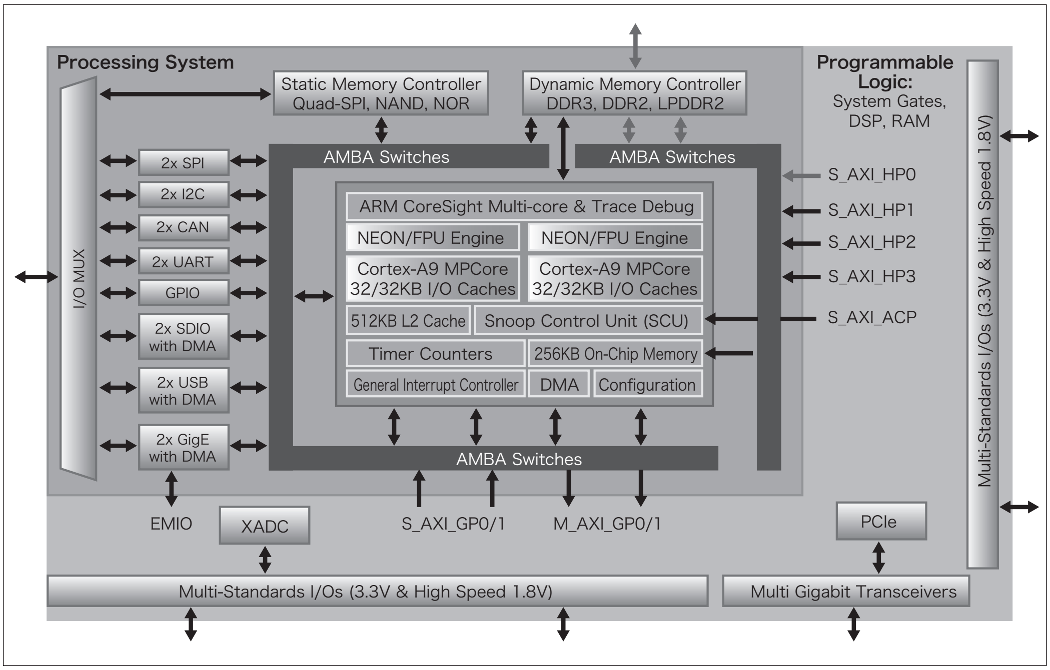
原因1　FPGA由数字电路、硬件发展而来

硬件的知识，特别是数字电路的设计技术，是非常必要的。必须要有设计工具和硬件的知识才能够设计FPGA。最近有了硬件描述语言（VHDL或者Verilog），因此可以用与C语言相似的语法为FPGA编程。但是，即使语法相似，如果开发者没有硬件的意识，还是没有办法很好地编程。有时还会遇到虽然描述相同，但编译却会出错这样的问题。

原因2　元件很贵

虽说最近FPGA的元件已经便宜了很多，但是如果对个人电脑速度没有特殊要求的话，个人电脑的价格还是要比FPGA便宜。

▼图2　Zynq的构架图



原因3　不知道FPGA的用途

在工作中，很多人还是不知道FPGA到底能用做什么，而且有些地方还比较难以理解，这也是FPGA得不到应用的一个原因。

Zynq系列的新FPGA

Zynq是由Xilinx公司发布的新款FPGA，它的特点是，搭载了双核ARM Cortex-A9处理器和处理器专用的I/O，加入了Xilinx最新的7系列FPGA，并且实现了高速信道互联（图2）。

因为搭载了Cortex-A9，自然就可以运行Android系统。CPU搭载了内存和其他接口。Cortex-A9双核的最大运行速度为1GHz，而且内置了高速缓存。平常个人PC所具有的东西都具备了，因此用户能够马上使用。搭载的FPGA为最新的7系列，其特点是耗电低，并且内置了A/D转换器。

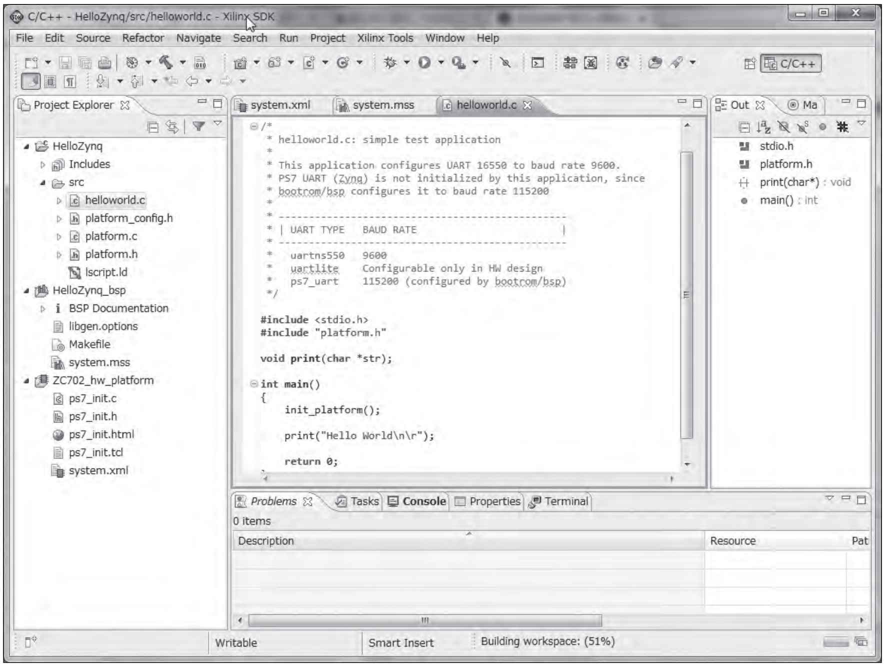
Zynq推翻了至今为止我们对FPGA的认识（也推翻了对CPU的认识），可以说是搭载了多种机能的一种装置，让我们从硬件和软件两方面，来看看它的特点。

硬件方面的特点

一般来说CPU+FPGA的组合，一直是以FPGA内嵌软核处理器，或者由CPU+FPGA两个芯片来构成的。Zynq不一样，其最明显的特征是，CPU和FPGA之间由一个叫做AXI的高速信道进行连接。并且无论是CPU还是FPGA都可以自由自在地访问内存。

一直以来，CPU向FPGA传输数据时，都需要增加数据传输用的程序和相应的硬件，还

▼图3　软件开发工具



需要使用其他方法对增加的部分进行检证。而Zynq里CPU把必要的处理保存在内存上，FPGA利用CPU传过来的内存地址，就可以访问存储器上的数据了。

软件方面的特点

以前说起FPGA的程序，就必须要学习硬件描述语言等数字电路的知识，Zynq里FPGA的设计可以用C语言来进行（不过开发工具还不完善，笔者试了一下，还有一些地方运行欠佳）。因为采用了这个技术，就不需要区分把哪里设计成软件、哪里设计成硬件、而是直接以软件开发的形式进行开发、调试，然后把需要高速处理的地方进行硬件化处理就可以了。

Zynq的新设计方法

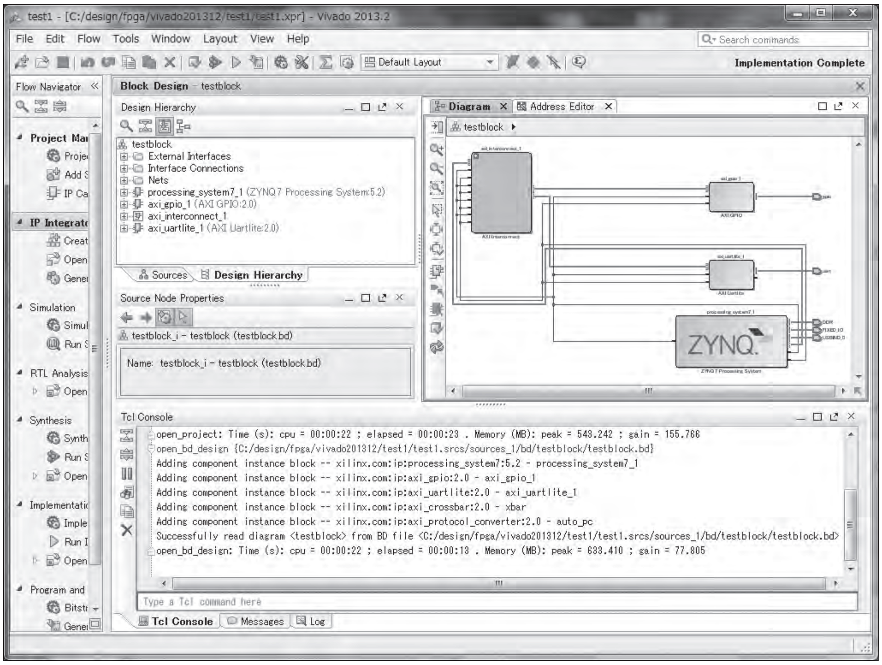
Zynq的设计方法有很多。和以前一样，有以硬件描述语言为基础，设定Cortex-A9和程序等方法。这些方法在FPGA的入门资料中都有介绍，这里就不再说明了。这里介绍一种还未普及的设计方法。

目前，Xilinx公司提供了几种开发工具。在这些开发工具当中，多数还是用以下三个工具进行设计的。

① 应用程序开发工具——SDK

一说到SDK，大家就会明白，这是一种软件开发工具（图3），可以应用在Cortex-A9的软件开发。它是以Eclipse为基础做成的统合环境，可以用C或者C++来开发。有Eclipse使用经验的人，很快就可以掌握。

▼图4　Vivado 2013.2



② FPGA的开发工具——Vivado 2013.2

Vivado为FPGA综合开发工具，也是Xilinx公司的主打工具（图4），它本来是可以对FPGA做很多细微调整的多功能工具，但因其使用C语言来做成FPGA，所以成了专门管理接线信息的工具。虽然使用Vivado多少需要学一些新东西，但使用Vivado可以省去对硬件的操作。

③C语言逻辑合成工具——Vivado HLS 2013.2

请看图5，这也是一种以Eclipse为基础的工具。这个工具是用来设计FPGA逻辑部分的。现在的FPGA设计，可以把需要的部分，从C语言转化成硬件处理。C语言做的程序中，可以把需要高速运行的部分，拷贝到HLS，把一部分描述到硬件。这样增加一部分程序之后，比CPU的处理快100倍的程序就完成了。

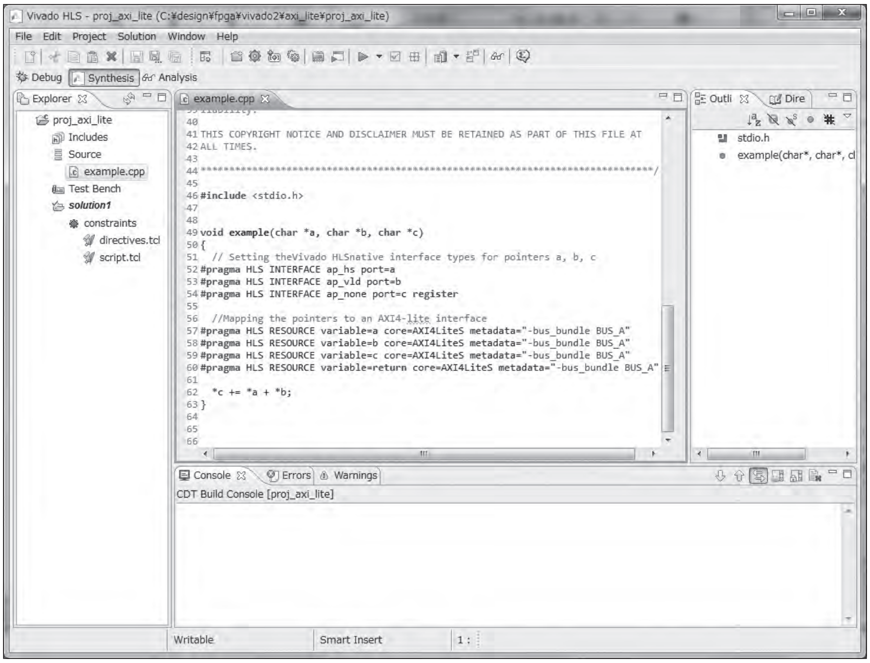
设计的流程是，用SDK①以软件制造的方式进行开发，把其中需要高速运行的部分拷贝到HLS③里硬件化。用Vivado②进行调试，使FPGA可以运行。

上述的设计工具，还在发展阶段，还并不具备统一性，多多少少都需要一些调整，但是和现在依赖硬件描述语言的主流设计方式比起来，可以更轻松、更快地进行设计。

在Android里使用Zynq的优势

Zynq的特点是，比平常软件的处理速度要快数十倍、甚至百倍。现在的智能手机，如果能搭载Zynq就太好了，不过这肯定不会那么快就能实现。目前的情况是，只要能在智能手机得到应用，就很引人瞩目了。不过，Android自身不仅针对智能手机，其他机器也可以用。智能手机的普及，带动了触摸屏等接口的普

▼图5　Vivado　HLS 2013.2



及，所以其他的电子设备也可搭载触摸屏和其他接口。基于这种状况，只要在Zynq上搭载了Android，就可以更加便捷地控制各种接口。

反过来，从Android角度来看，如果想更加快速，FPGA是一个不错的选择。最近，对于图像和大数据的处理越来越多了。只有CPU的话，处理可能力不从心。而只要把程序的一部分FPGA化，速度就会大幅加快。

智能手机中搭载FPGA还有很长的一段路要走，但如果能在Android手机里搭载Zynq，即使想要随着时代发展更改通信传输方式（LTE->LTEAdvance等），也仅仅需要更新软件就可以了，那该多好啊。

像Zynq这种CPU+FPGA的形式很吸引人的眼球，很多厂商都有这方面的制造计划。程序的性能优化是永远的课题，而像Zynq这样的装置如果能成为性能优化的助力就再好不过了。

小山忠昭　FPGA信息有限公司　法人

1987年开始接触于1985年问世的FPGA，从此便与FPGA结缘，直至今日。现在，主要从事FPGA的设计、开发工作，也兼职Xilinx的培训讲师。日本的安卓协会神户支部（主要是硬件方面）成员。

如何构建超级系统管理程序 Hypervisor

正确理解虚拟化技术

第11回 基于virtio的半虚拟化设备之一 virtio概要与Virtio PCI

文/浅田 拓也（ASADA Takuya）Twitter @syuu1228　译/honnkyou

开　始

到上回为止，我们以BHyVe源代码为例对超级系统管理程序的I/O虚拟化安装进行了解说。本回，我们讲解一下能够大幅度提升客户机OS I/O性能的virtio半虚拟化设备的基本情况以及virtio组件之一的Virtio PCI。

全虚拟化设备与半虚拟化设备

在虚拟x86结构的方法中有一种方式称为半虚拟化，该方式是基于Xen实现的。

半虚拟化过程中为适应虚拟化要求，会对客户机操作系统做一些适当的修改。因此，与不修改客户机操作系统的全虚拟化相比，能够获得更高的性能。而且，半虚拟化也不需要硬件虚拟化的支持。

现阶段，与半虚拟化相比，全虚拟化的应用更加广泛。这是因为具有硬件虚拟化支持功能的CPU得到了普及，在很多环境下都可以使用高速硬件支持功能。

但是，在采用全虚拟化的超级系统管理程序中也引入了一部分半虚拟化概念。本期我们要讲解的，就是这部分中对整个系统性能有较大影响的一些虚拟设备。

这类虚拟设备称为半虚拟化设备。与之相对，模拟与真实硬件设备相同的虚拟设备称为全虚拟化设备。

全虚拟化设备可以直接使用操作系统中适用于真实硬件的设备驱动，但半虚拟化设备则需要安装适用于客户机环境、半虚拟化设备专用的设备驱动。

半虚拟化设备中有virtio框架，这是一种不依赖特定管理程序或操作系统的框架。其软件规格和源代码都已开源，因此超级系统管理程序中的KVM、VirtualBox、lguest、BHyVe等，客户机操作系统中的Linux、FreeBSD、NetBSD、OpenBSD、Windows、MonaOS等很多系统都将其作为标配预先安装了。

另外，Xen和VMware、Hyper-V等超级系统管理程序也采用了方式相同的半虚拟化驱动[①]。

全虚拟化设备慢的原因

在实际机器上对网格接口和块设备进行I/O操作时，操作系统以设备驱动为媒介对各种设备的硬件寄存器进行读写操作。

① 与virtio不同的独立设备也在使用。

如前几回讲述的一样，Intel VT-x每次检测相应的硬件寄存器访问时都执行VMExit指令。超级系统管理程序接收VMExit指令，进行设备模拟处理。

这一系列的处理是只在进行虚拟化时发生的系统开销，这部分的处理效率体现在与实际机器比较时I/O性能的差别上。

更详细地说，是会发生如下开销，与实际机器上的I/O相比，会有增加延迟的可能性。

VMX non-root mode与VMX root mode间模式转换命令的开销

访问硬件寄存器时的VMExit指令与客户机再启动时的VMEntry指令，各自在VMX non-root mode和VMX root mode模式间进行模式转换。这种模式转换开销虽然随着CPU升级在逐渐变小，但VMExit与VMEntry指令也要消耗1000周左右。

调用设备模拟程序命令的开销

在多数情况下，超级系统管理程序的设备模拟程序运行在用户进程上。因此，在模拟硬件寄存器访问的过程中需要由内核模式向用户模式迁移，模拟结束后则需要返回到内核模式。

另外，用户进程要在进程调度器判断为合适的时间执行，因此无法保证VMExit指令结束后能够立即执行设备模拟程序。

同样，客户机再启动的VMEntry指令也不能保证在设备模拟结束后能立即被执行，也有可能转到挂起状态等待调度。

再有，一般的全虚拟化设备一次I/O调用需要执行多次寄存器访问(例如，某型号网卡的一次接收处理需要5～6次的寄存器访问)。每次执行寄存器访问时都会发生上述模式转换，需要很大的开销。在需要高速I/O设备的情况时，这点就会成为性能上的瓶颈。

virtio概要

virtio为了减少上文所述的设备全虚拟化过程中的开销，在宿主机与客户机之间通过放置在共享内存中的队列进行数据读写。

VMExit向队列中送出数据时，只是为了向超级系统管理程序发出通知，并且如果此时超级系统管理程序正在处理队列中的数据，通知可能会被延迟。因此，与全虚拟化设备相比，virtio大幅度减少了模式转换的次数。

virtio大致可分为Virtio PCI和Virtqueue两个部分。

Virtio PCI对于客户机来说相当于PCI设备，提供以下功能。

- 设备初始化时的宿主机与客户机间协商或通知设定信息时使用的配置寄存器。
- 中断(宿主机→客户机)、I/O端口访问(客户机→宿主机)等引起的宿主机与客户机之间的事件通知功能。
- 使用标准的PCI设备DMA进行数据转送的功能。

Virtqueue是数据转送时使用的客户机内存空间中的队列结构。每个设备可以拥有一个或者多个队列。例如，virtio-net需要送信队列、收信队列和控制队列三个队列。

客户机操作系统检测到作为PCI设备的virtio设备后对其进行初始化，向Virtqueue中存取数据，中断和I/O访问则通过事件通知方式对宿主机提交I/O申请。

趁此机会，我们再来认真研究一下Virtio PCI的相关内容。

回顾PCI

在进行Virtio PCI讲解之前，我们首先简单回顾一下PCI的相关知识。

PCI设备通过Bus Number和Device Number来识别，一个设备具备多种功能时，则由Function Number区分各个功能。

这些设备占有PCI Configuration Space、PCI I/O Space、PCI Memory Space三个内存空间。

PCI Configuration Space包含了大量设备初始化和驱动载入时需要的信息。其中有显示设备厂商代码和设备号的VendorID和Device ID信息；显示PCI I/O Space和PCI Memory Space映射地址的Base AddressRegister信息和MSI中断设定信息等。

要想访问PCI Configuration Space需要按照下面的步骤执行。

① 将设备的Bus Number，Device Number，Function Number和要访问领域的offset值连同Enable Bit一起设定到CONFIG_ADDRESS寄存器[②]中。CONFIG_ADDRESS寄存器的位配置如表1所示。

② 对CONFIG_DATA寄存器[③]执行读入或写入操作。

操作系统在PCI设备初始化时，以迭代Bus Number、Device Number后按顺序访问PCI Configuration Space的方式，就能检测出连接到计算机上的PCI设备。

PCI I/O Space映射在I/O空间里，主要用于设备的硬件寄存器映射等[④]。

如图1所示，PCI Memory Space映射在物理地址空间内，使用在需要如图像存储器等大内存空间的情况下。无论哪个领域映射地址都需要参照PCI Configuration Space的Base Address Register取得。

Virtio PCI设备检测方法

virtio设备是作为连接到客户机上的PCI设备被客户机操作系统识别的。

这时，PCI Configuration Space的Vendor ID为0x1AF4，Device ID为0x1000 - 0x1040的值。更进一步，为了判别virtio设备种类，还会配有如表2所示的Subsystem Device ID作为追加信息。

客户机操作系统查看这些ID后就会载入适当的virtio驱动。

▼ 表1　CONFIG_ADDRESS register

位(bit)	名称(name)
31	Enable Bit
30–24	Reserved
23–18	Bus Number
15–11	Device Number
10–8	Function Number
7–2	Register offset
1–0	0

▼ 图1　PCI Configuration Space

31–24	23–16	15–8	7–0	
Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
BIST	Header Type	Lat. Timer	Cache Line S.	0Ch
Base Address Registers				10h, 14h, 18h, 1Ch, 20h, 24h
Cardbus CIS Pointer				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Expansion ROM Base Address				30h
Reserved			Cap. Pointer	34h
Reserved				38h
Max Lat.	Min Gnt.	Interrupt Pin	Interrupt Line	3Ch

② PC机的CONFIG_ADDRESS寄存器映射到I/O空间的0xCF8处。
③ PC机的CONFIG_DATA寄存器映射到I/O空间的0xCFC处。
④ PC机上是映射到I/O空间中，在其他架构的机器上也有映射到内存的情况。

Virtio Header

Virtio Header是设置在PCI I/O Space开始处的，用于设定virtio设备的字段，客户机操作系统在virtio设备驱动初始化时使用Virtio Header（表3）。

Virtio Header的尾部（MSI无效的情况是20byte，有效的情况是24byte）接着device specific header。

表4中列出了device specific header。

▼表2　Subsystem Device ID

子系统设备标识（Subsystem Device ID）	设备类型（device type）
1	network card
2	block device
3	console
4	entropy source
5	memory balooning
8	SCSI host
9	9P transport

Virtio PCI设备的初始化处理

采用客户机操作系统中的Virtio PCI进行virtio设备初始化处理的步骤如下所示。

Step 1

执行日常PCI设备初始化例程，检测virtio的Vendor ID和Device ID。

Step 2

获取设备的PCI I/O Space的映射地址，在Virtio Header的STATUS字段中设置ACKNOWLEDGE位（STATUS字段使用的位值在表5中记载）。

Step 3

加载与Subsystem Device ID一致的驱动。例如，ID为1时加载virtio-net。

▼表3　Virtio header

偏移量（offset）	字段名（field name）	字节（bytes）	配置方式（direction）	说明（description）
0	HOST_FEATURES	4	RO	宿主机提供的功能的位字段
4	GUEST_FEATURES	4	RW	客户机想要实现的功能的位字段
8	QUEUE_PFN	4	RW	分配给QUEUE_SEL指定队列的内存物理页的页帧号（PFN）
12	QUEUE_NUM	2	RO	QUEUE_SEL指定的队列大小
14	QUEUE_SEL	2	RW	队列号
16	QUEUE_NOTIFY	2	RW	用于通知QUEUE_SEL指定队列中有应该处理的数据
18	STATUS	1	RW	设备的状态
19	ISR	1	RO	中断状态
20	MSI_CONFIG_VECTOR	2	RW	控制队列的MSI向量号（只有MSI有效时存在）
22	MSI_QUEUE_VECTOR	2	RW	QUEUE_SEL指定队列的MSI向量号（只有MSI有效时存在）

▼表4　virtio-net specific header

偏移量（offset）	字段名（field name）	字节（bytes）	配置方式（direction）	说明（description）
0	MAC	6	RW	MAC地址
6	STATUS	2	RO	Link up等virtio-net固有状态
8	MAX_VIRTQUEUE_PAIRS	2	RO	最大RX/TX队列数（用于多队列）

◀ **Step 4**

驱动加载成功的话，在Virtio Header的STATUS字段中设置DRIVER位。

◀ **Step 5**

执行设备固有的初始化处理。virtio-net的情况下，从virtio-net specific header中拷贝MAC地址，作为网卡进行向网络子系统注册等处理。此时，在Virtio Header的HOST_FEATURES字段中显示的设备的可用功能中，将驱动想使用的功能的位写入到GUEST_FEATURES中。这里不再赘述FEATURES的所有位值，例如virtio-net想要使用Checksum Offloading时位值为0，想要使用TSOv4时必须设置位值为11。

◀ **Step 6**

分配设备所需数量的队列，通过Virtio Header向宿主机发出地址通知（在“队列的分配处理”中详细介绍）。

◀ **Step 7**

驱动的初始化处理全部成功的话，在Virtio Header的STATUS字段中设置DRIVER_OK位，中途失败的话设置FAILED位。

队列的分配处理

“Virtio PCI设备的初始化处理”中Step 6提到的队列的分配处理需要在每个队列中执行如下的步骤。virtio-net时需要3个队列所以就需要重复3次。

◀ **Step 1**

将执行设定的队列号写入到Virtio Header的QUEUE_SEL字段中。

◀ **Step 2**

读取Virtio Header的QUEUE_NUM字段。该字段值是要执行设定的队列的队列长。值为0时，表示宿主机不同意使用该号码对应的队列，所以不能使用该队列。

◀ **Step 3**

分配队列需要使用的内存空间。分配的大小需要符合队列长度，开始地址需要匹配分页大小。

◀ **Step 4**

将内存空间开始地址的页帧号设置到Virtio Header的QUEUE_PFN中。

◀ **Step 5**

MSI中断有效时，向Virtio Header的MSI_CONFIG_VECTIOR字段或者MSI_QUEUE_VECTOR字段中写入中断向量号。究竟写入到哪个字段，根据队列是否是控制队列而有所不同。

▼表5　device status

位（bit）	名称（name）	说明（description）
1	ACKNOWLEDGE	客户机OS发现设备
2	DRIVER	客户机OS持有设备对应的驱动
3	DRIVER_OK	客户机OS完成驱动初始化
7	FAILED	初始化失败

结束语

本回我们讲解了virtio概要和关于Virtio PCI安装的内容，下一回则要讲解Virtqueue和使用Virtqueue的网卡（virtio-net）的实现方法。

Universal shell programming 研究所　http://www.usp-lab.com
上田隆一（UEDA Ryuichi） Twitter @uecinfo　译 / honnkyou

第20回 编写CGI脚本（2）——使用GET方法获取字符串

前　言

又到了每期大彻大悟Shell脚本连载的时间。上一回说到用bash执行CGI脚本，其实这类禁止技在过去并不十分罕见，与Unix的老用户谈到这一话题时，大家都相当怀念。

但被怀念并不是一件值得高兴的事，本连载也不是为了追忆古老的技术。但是，如果掌握了用bash编写CGI脚本的技能，就可以在本地编写CGI再上传到服务器上，还是十分方便的。为了不使这一传统技能消失，笔者决定在这里偷偷记录下这一方法。

这一回，我们研究的课题是，使用GET方法通过浏览器向CGI脚本传送字段，服务器端CGI脚本动态显示接收到的文字。

gets！——Dandy坂野

环　境

书接前回，笔者使用手中的Mac启动Apache服务，并配置解析CGI脚本。Mac下Apache的设定方法请参照上一回。Linux、BSD的设定方法网上参考资料很多，请自行查找设定。

上一回已经介绍过，笔者的Mac是OS X标准版，运行的是自带安装的Apache。将编写的CGI脚本放到路径/Library/WebServer/CGIExecutables/下，打开浏览器输入地址http://localhost/cgi-bin/hoge.cgi，页面能正常打开说明配置成功，CGI脚本可以正常启动。上一回也是这样设定的，不过将文件一个一个移到/Library/WebServer/CGIExecutables/文件夹下是一件非常麻烦的事。所以如图1所示，笔者在自己用户的主目录下创建了名为cgi-bin的符号链接，连接到/Library/WebServer/CGIExecutables/文件夹，这样只要将脚本放到cgi-bin下即可。

本回内容对版本没有过多要求，不过为了以防万一仍需确认一下Mac的软件环境是否如图2所示。

GET使用方法

首先从最简单的GET讲起。GET（GET方法）是在HTTP中向CGI脚本传递字符串的方法之一。做法是在浏览器地址栏中输入URL时，将字符串紧贴在地址后面，然后将字符串传递给CGI 。

▼图1　轻松从主目录访问脚本

```
$ ln -s /Library/WebServer/CGIExecutables/ ./cgi-bin
$ cd cgi-bin
$ sudo chown ueda:wheel ./
```

▼图2　环境

```
$ bash --version
GNU bash, version 3.2.48(1)-release (x86_64-apple-darwin12)
Copyright (C) 2007 Free Software Foundation, Inc.
$ uname -a
Darwin uedamac.local 12.3.0 Darwin Kernel Version 12.3.0: Sun Jan  6 22:37:10 PST 2013;
root:xnu-2050.22.13~1/RELEASE_X86_64 x86_64
$ apachectl -v
Server version: Apache/2.2.22 (Unix)
Server built:   Dec  9 2012 18:57:18
```

代码清单1中通过shell脚本展示了一个实例。将shell脚本保存至cgi-bin下，执行chmod+x命令赋予任意用户可执行权限后，通过浏览器运行该脚本，显示结果如图3所示。

在这里稍微解释一下，在地址栏中输入的字符串是http://localhost/cgi-bin/echo.cgi?gets!!它的意思就是通过GET向echo.cgi脚本传递gets!!这一字符串。

被传递字符串的CGI脚本必须用某种方法接收传递过来的字符串。方法很简单，就是给传递过来的字符串添加QUERY_STRING变量并使用该变量即可。所以，如代码清单1那样添加HTTP协议头，执行简单的echo命令，就可以将通过GET传递过来的字符串输出到浏览器中。

小心注入

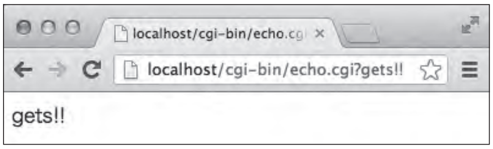
使用变量QUERY_STRING时，只要没有特殊情况，就要像第5行那样将变量用双引号括起来。如果不括起来的话就可能发生如图4那样的情况，这与在终端上的规则相同。如图5所示那样在终端上做一下实验，应该可以理解上述情况。

▼代码清单1　表示使用GET方法获取字符串的CGI脚本（echo.cgi）

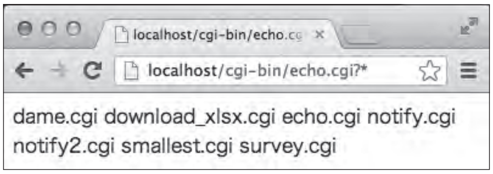
```
01 #!/bin/bash -xv
02
03 echo "Content-type:text/html"
04 echo
05 echo "$QUERY_STRING"
```

有一点要记住，无论怎样，使用shell脚本编写CGI脚本都跟系统有密切的联系。但是也不用过度担心被注入恶意命令。即使没有用引号括起$QUERY_STRING变量，echo后紧跟的变量也只是被转化为字符串而不会被执行（图6）。

▼图3　使用GET方法传送字符串



▼图4　QUERY_STRING不使用双引号则传送“*”



▼图5　在终端上实验有无括号

```
将[*]设置值为变量A
$ A="*"
不用括号括起来
$ echo $A
dame.cgi download_xlsx.cgi（略）
用括号括起来
$ echo "$A"
*
```

▼图6　分号后注入命令



相反，如图7格式书写代码的话，则会发生意想不到的的问题。如例所示，即使用引号括起也不是绝对安全的。

▼图7　GET接收的命令字串被执行实例

```
 1 变量放行首 
$ cat yabai1.cgi
#!/bin/bash -xv

echo "Content-type: text/html"
echo
"$QUERY_STRING"

 命令可执行 
$ curl http://localhost/cgi-bin/yabai1.
cgi?ls
dame.cgi
download_xlsx.cgi
echo.cgi
（以下略）

 2 使用eval 
$ cat yabai2.cgi
#!/bin/bash -xv
echo "Content-type: text/html"
echo
eval "$QUERY_STRING"

 命令可执行 
$ curl http://localhost/cgi-bin/yabai2.
cgi?ls
dame.cgi
download_xlsx.cgi
echo.cgi
（以下略）
```

还有其他一些不好的书写格式，但这里我们简单了解这两种类型并在编码中注意就不会有太大问题。当然，在封闭的内部环境中实验时没有必要考虑这么详细。笔者一直认为安全级别应根据网站用途的来设定，设定后我们进行下一步。

选择命令展现执行结果

接下来让我们使用GET实践一下。这里我们要编写的是一个服务器管理页面，以及可以通过页面直接调用命令的CGI脚本。这之前我们也编写过输出HTML的脚本，但本回我们不是事先在脚本中写入HTML语句，而是让CGI动态生成HTML语句。

首先，用编辑器创建一个如代码清单2所示的html文件。用CGI读取数据，再由sed等命令加工数据后动态生成HTML语句。

接下来，准备好如代码清单3所示的CGI脚本文件，表示上一步创建的html文件。为了调试，我们在后面添加了echo "$QUERY_STRING"命令。生成html语句后才输出Debug信息有取巧

▼代码清单2　com.html

```
01 <!DOCTYPE html>
02 <html>
03     <head>
04         <meta charset="UTF-8" />
05         <title>cgi test</title>
06     </head>
07     <body>
08         <form name="FORM" method="GET" action="./com.cgi">
09             命令:
10             <select name="COM">
11                 <option value="0">cat /etc/hosts</option>
12                 <option value="1">top -l 1</option>
13             </select>
14             <input type="submit" value="执行"/>
15         </form>
16         <pre>
17             <!--RESULT-->
18         </pre>
19     </body>
20 </html>
```

之嫌，但这样做Debug信息就可以显示在浏览器页面中。

接下来用浏览器调用com.cgi文件，从下拉选框中选中一个项目后按下确认按钮。如图8所示左下角就会显示GET方法传递过来的字符串。

HTML语句中嵌入列表

接下来我们来解释一下图8中的“COM=1”。COM字符串是下拉列表选框的名字（com.html中name="COM" 的部分），“=”右边则是选择项目的value值。

获取GET字符串的CGI脚本通过value值可以知道浏览器端选择了哪个选项，因此直接执行下拉列表选框中记录的命令就可以了。编号与命令对应表的文件应该放在哪里比较好呢？而且，虽然直接将命令写在com.html文件中了，但对应表文件的内容还是动态实现比较好吧？

这个时候，让我们使用open usp Tukubai（https://uec.usp-lab.com）的mojihame命令。首先把com.html替换为代码清单4中的内容。

接下来用代码清单5中的内容替换com.cgi。

▼代码清单3　表示从下拉选框中所选项目的CGI脚本（com.cgi）

```
01 #!/bin/bash -xv
02
03 htmlfile=/Users/ueda/cgi-bin/com.html
04
05 ### 显示内容
06 echo "Content-type: text/html"
07 echo
08 cat $htmlfile
09
10 # 调试代码
11 echo "$QUERY_STRING"
```

▼代码清单4　对应了mojihame的com.html

```
01 <select name="COM">
02 <!-COMLIST-->
03     <option value="%1">%2</option>
04 <!-COMLIST-->
05 </select>
```

这样在浏览器中写到$tmp-清单中的命令就会加上编号（行号）并显示到下拉列表中。命令列表也可以是外部文件，不过这里为了演示使用了嵌入文档。

我们接下来要说明的是跟本回主题没有关系的几个细节，代码清单5中第一行的“-xv”是脚本执行时输出日志的选项。第3行的“exec 2>”是用来重定向脚本错误信息到文件中的命令。第10行到14行的嵌入文档的作用是将两个“FIN”之间的代码输出到标准输出中。其中“FIN”只要是开始与结尾成对出现，那么也可以换成EOF或者HOGE。

▼图8　表格传送的字串



▼代码清单5　将命令列表嵌入com.html的com.cgi

```
01 #!/bin/bash -xv
02
03 exec 2> /tmp/log
04
05 PATH=/usr/local/bin:$PATH
06
07 htmlfile=/Users/ueda/cgi-bin/com.html
08.tmp=/tmp/$$
09
10 cat <<FIN > $tmp-list
11 cat /etc/hosts
12 top -l 1
13 echo test_test_
14 FIN
15
16 ### 显示内容
17 echo "Content-type: text/html"
18 echo
19 sed 's/_/\\_g' $tmp-list    |
20 tr ' ' '_'           |
21 awk '{print NR,$1'        |
22 mojihame -lCOMLIST $htmlfile -
23
24 # 调试代码
25 echo "$QUERY_STRING"
26
27 rm -f $tmp-*
28 exit 0
```

如果不考虑mojihame部分，首先，第21行awk命令后面的管道“|”中保存的是如代码清单6所示的数据。添加行号，将空格转变成“_”，在原本的“_”前加转义符“\”。open usp Tukubai命令因为接收以空格分隔的数据，因此不得不将数据转换一下格式。

这样就会有两列数据（一列数据是下拉列表中的value值，另一列是下拉选框中的命令）。将这两列数据写入mojihama后，COM清单括起来的部分只复制记录数，第1列的内容赋值给代码清单4中的%1，第2列的内容赋值给代码清单4中的%2。被转码的内容还原为原来形式。Mojihama输出的HTML中，下拉选框的部分表示为代码清单7的样子。

熟悉mojihama后你会觉得它很方便。不怕麻烦的话，只用awk也可以写出HTML页面。

再次小心注入

这里我们需要再注意一件事，com.cgi从下拉选框获取数字，但并不是只能获取数值数据。如图9所示，使用curl等命令，或者在浏览器地址栏中修改URL后带的字符串也可以向服务器端输送危险的字符串。

这种情况的应对方法相对复杂，不过我们的例子只是单纯地接收数值，所以接收到GET命令传过来的数据后会立即用tr命令删除非数值字符。在`tmp=/tmp/$$`行的下一行追加下面一行代码。

```
NUM=$(echo "$QUERY_STRING" |tr -dc '0-9')
```

tr参数“`-d`”的作用是删除字符，参数“`-c`”则是匹配字符。追加后的结果如图10中所示（UTF-8编码的话即使混入非数值字符也没问题）。这样下来只有12保留，甚至连换行符都消失了（行尾的提示符$就是为了去除换行符）。这样行号就赋值给变量NUM，剩下的就是执行列表中的命令。

完　成

完成的com.cgi如代码清单8所示。字符串值传不到变量中、中间文件创建不成功等Bug会时有发生，需要稍微适应一下。例如，如果命令输入不到COM中，在第23行的位置就会报错。

▼图9　com.cgi 中直接用GET 传递数据

```
$ curl "http://localhost/cgi-bin/com.cgi?reboot"
（略）
</html>
reboot
```

▼图10　用tr命令删除指定外文字

```
$ echo 'COM=1aewag啊啊 2' | tr -dc '0-9'
12$
 ↑ 只保留1和2
```

▼代码清单6　换码后的命令列表

```
01 cat_/etc/hosts
02 top_-l_1
03 echo_test\_test_\_
```

▼代码清单7　com.cgi执行后生成的部分html代码（下拉列表部分）

```
<select name="COM">
    <option value="1">cat /etc/hosts</option>
    <option value="2">top -l 1</option>
    <option value="3">echo test_test_</option>
</select>
```

所以在第21行的位置添加“:”（无操作命令）等细节技巧是很有必要的。但是，如果行数相对较少，也能让大家尽快习惯，然后就可以试着开发点什么，或者可以早日制作出USP同好会网站（http://www.usptomo.com）这样美观的网站。

对于成品，还需要使用一个名为filehame的命令。filehame命令的作用是将某一文件内容插入到另一文件中，使用方法如图11所示。这里用sed命令也可以达到相同的效果。

▼代码清单8　调用命令的CGI（com.cgi 最终版）

```
01 #!/bin/bash -xv
02
03 exec 2> /tmp/log
04
05 PATH=/usr/local/bin:$PATH
06
07 htmlfile=/Users/ueda/cgi-bin/com.html
08 tmp=/tmp/$$
09
10 ####### 可以执行的命令列表 #######
11 cat <<FIN > $tmp-list
12 cat /etc/hosts
13 top -l 1
14 echo test_test_
15 FIN
16
17 ####### 命令执行 #######
18 #获取编号
19 NUM=$(echo "QUERY_STRING"|tr -dc '0-9')
20 #获取指定行内容
21 COM=$(awk -v n="$NUM" 'NR==n' $tmp-list)
22 #COM内容为空时，用":" 代替
23 [ -z "$COM" ] && COM=":"
24 #执行
25 $COM > $tmp-result
26
27 ####### 生成HTML 代码 #######
28 echo "Content-type: text/html"
29 echo
30 #换码
31 sed 's/_/\\_g' $tmp-list    |
32 tr '' '_'              |
33 # 加编号
34 awk '{print NR,$1'         |
35 #输出 >>> 1:行号 2: 命令
36 mojihame -lCOMLIST $htmlfile - |
37 #嵌入命令执行结果
38 filehame -lRESULT - $tmp-result
39
40 rm -f $tmp-*
41 exit 0
```

最后的执行结果如图12所示。无论选择哪个命令，按下确认按钮，下拉选框的选择结果都会返回缺省值“cat /etc/hosts/”，不过根据命令可能会不同。这里使用了open usp Tukubai的formhame命令，对于这个命令有机会我再讲解。

结　束

本回我们介绍了使用GET方法向CGI脚本传送字符的方法。制作了一个由浏览器执行命令的应用程序。com.html和com.cgi合起来也就是60行左右的代码，经常在终端敲代码或编写shell脚本的人一定能够牢记于心，这样的话在需要的时候它就能发挥用武之地了。

▼图11　filename 的使用方法

```
$ cat file1
===参加者===
ATT
===以上===
$ cat meibo
山田
里中
殿间
$ filehame -lATT file1 meibo
===参加者===
山田
里中
殿间
===以上===
```

▼图12　执行结果



可以使用SDN吗？应该使用SDN吗？

网络虚拟化的陷阱

Data Hotel股份有限公司
伊势幸一（ise kouichi）
Twitter @ibucho
译/苏祎

第3回

检验OpenFlow

这一回是这本连载的最终回。最后我们来研究一下OpenFlow。如上回所述，若网络虚拟化的目的是形成不依赖于物理网络的叠加段和扩大IEEE802.1Q中的VLAN上限，那么只通过OpenFlow交换协议是不能实现网络虚拟化的。因此，虽然严格来说OpenFlow并不在本连载所包含的网络虚拟化的范围内，但使用流（flow）来驱动网络这样的架构，可以实现非常接近于网络虚拟化的状态，所以，笔者就尝试以OpenFlow作为本回讨论的对象。

想必很多人对OpenFlow还没有清晰的概念，还不清楚其到底能解决什么样的问题、适用于怎样的环境。在这种情况下，就在考虑是否引入OpenFlow。要想理解那些因为厂商鼓吹的积极印象而使本质变得模糊的革命性技术，不能仅了解它被广泛宣传的长处和优越性，还需要找出它的缺陷，这样才有更好的效果。了解这项技术无法解决的问题，反而能突出其适用范围。使用标榜了具有优越性和先进性的技术并以此为前提来讨论用途用法，容易产生错误的设计和运用，很可能会陷入纯粹浪费劳力和投资的后果。[①]

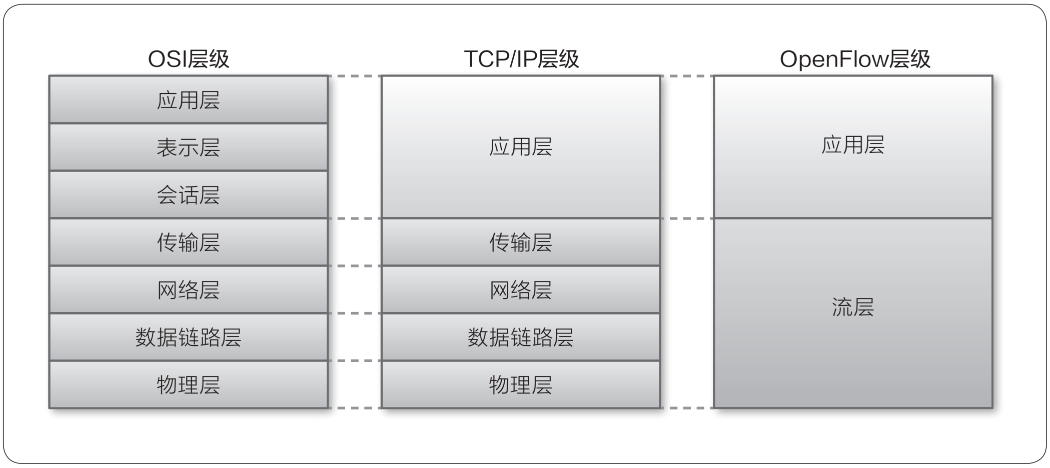
OpenFlow协议栈

关于普通的OpenFlow技术方面的内容，已经在各种文章和书籍中都有介绍，这里不再赘述。这里以已经理解了关于OpenFlow协议和交换，以及网络组成的基本技术及规范为前提进行说明。

有人评价OpenFlow是一种崭新的革命性系统，这种评价并不为过。以前，代表性的开放协议栈，有OSI协议和TCP/IP协议两种。而OpenFlow的协议栈，可以说是作为另一种层级存在的。但是也不能说它是完全新的协议层级模型，毕竟OpenFlow层级是把OSI层级的下面4层作为一个流层（flow layer）实现的。因此，

① 这是经验者之谈。

▼图1　开放式协议栈



在OpenFlow协议栈中，确定转发、更新、丢弃等行为的流标识符（flow identifier）不是IP地址和VLANID等单一的标识符，而是由TCP/IP协议栈中的下面4层组合而成的。并且，它不是由所有的下面4层组合而成，而是下面4层中任意标识符的组合，从而使OpenFlow能组成高自由度的网络拓扑。但从另一方面来说，这也成为了导致网络设计混乱的主要原因（图1）。

对OpenFlow的误解

听说有人批判OpenFlow，说它是对OSI和TCP/IP进行了层级侵犯，这是完全没道理的。本来，它和OSI、TCP/IP就是不同的协议栈，和别的栈里的层级模型相比，有不同的定义和处理是很正常的，如果这就被称为是层级侵犯，那么TCP/IP的应用层也在侵犯OSI的上面3层。

使用这个流进行转发，从表面来看，可以像在现有的物理网络上叠加一个流网络那样来使用。并且，如果这个流标识符的元素作为广播地址来使用还能形成虚拟段。若不考虑运行负载的话，共同通过VLAN-ID和虚拟机的MAC地址或IP地址的组合，能在虚拟环境下生成4094个以上的租户段。

这样只将OpenFlow理解为是TCP/IP的扩展协议或者是革命性的改进，就无法理解它的本质。必须意识到它是与以前的完全不同的新协议栈，使用这个栈设计运用的网络也将大为不同。

边端点和逐跳

OpenFlow中有两种部署方式：边端点（edge endpoint）和逐跳（hop-by-hop）。这两种部署方式也是导致混乱的主要原因。简单来说，边端点方式是以流驱动物理服务器内的网络，而逐跳是以流驱动物理服务器之间的网络。

但是，对于用数百台到数千台的物理服务器组成的云平台和数据中心，局限于驱动物理服务器内网络的边端点方式，基本上没有用处。

可是，只要服务器之间使用的是传统的物理网络，服务器之间的通信就不能使用流驱动，

因此OpenFlow边端点也就只能作为服务器内流网络和现有TCP/IP网络的网关来使用，不可能在网络整体形成流。

边端点方式

这种情况下，可以使用边端点方式把服务器之间的通信隧道化，在现有的物理网络上叠加流网络来对应。这样，就自然而然地再一次延续了上回提到的使用端点模型虚拟化引起的弱点和缺陷，可以说这是一个适合于相较于高速通信，更优先于段的灵活生成的网络协议（图2）。

逐跳方式

逐跳方式是一个以域内的所有物理交换机都能流驱动为前提的网络。服务器之间的路径节点都已经使用流驱动，路径并不需要隧道化，几乎没有网络叠加的性能损失。

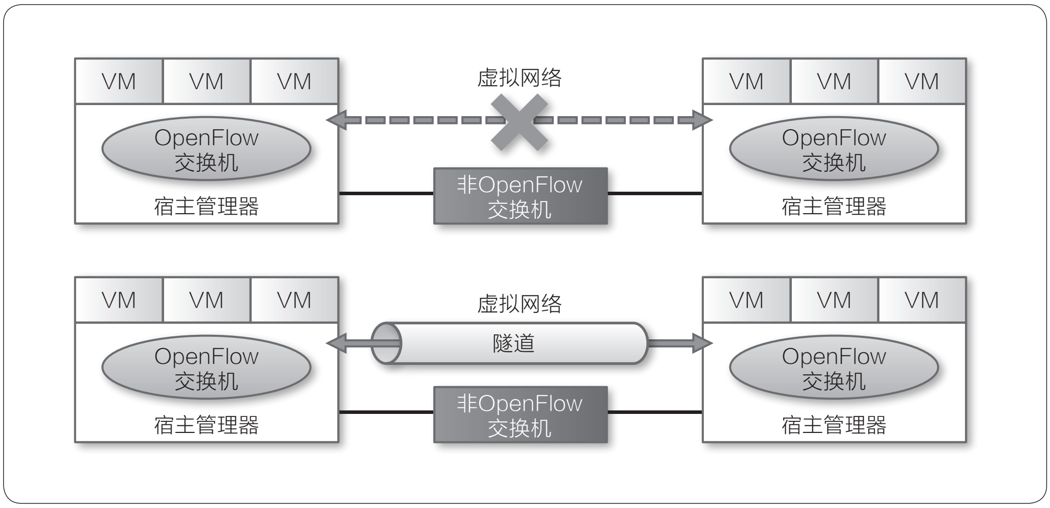
若要域内的所有交换机都支持OpenFlow，可能需要采购设备，这会有一定的经济方面的问题。但将OpenFlow的流驱动方式商品化多少能解决一部分问题。逐跳方式的情况下，因为服务器之间的通信路径能完全支持流驱动，所以完全不用考虑以前的分层级转发规则，而且仅仅使用流就能形成通信隧道逻辑。

使用流生成网络拓扑，与MPLS等传统叠加方式以及上一回讨论的虚拟网络相比，是完全不同的新网络架构。但是，实际上因为流是由TCP/IP下面四层组成的，所以从某种意义上来说，也可以说是现有网络的虚拟化或叠加。

实操中的错误与误解

因为边端点方式和逐跳方式在实现和功能上完全不同，最好还是不要将其视为相同架构。换句话说，在OpenFlow中有两种部署模型是导致混乱的原因，即端点模型和逐跳这两种网络架构，它们各自使用同称为OpenFlow的协议栈在运行，这样理解会更形象。但是，因为使用同称为OpenFlow的协议，所以可以通过将双方互相连接，形成服务器内VM之间和服务器之间的通信都用流来驱动的网络。这相当于，

▼图2　边端点模型



若所有组成网络的服务器、交换机、路由器都支持TCP/IP的话，就能各自互相连接，形成一个网络。

那么OpenFlow的致命缺陷是什么呢？主要有以下几个问题。

控制层的分离问题

潜在的网络不稳定性

首先，数据层和控制层是分离的。操作OpenFlow交换机的控制实例和交换机实例是不同的。分离数据层和控制层，目的在于使用一个控制器集中管理多个数据层。理想情况下，一个控制器可以管理域内所有的OpenFlow交换机。

但是，反过来说，若控制器和交换机的通信隧道有问题的话，网络的控制也会变得不稳定。交换机流表（flow table）的更新方式有两种：一种是主动方式，启动时下载在控制器端的流控制程序，生成流表；另一种是被动方式，如果接收到的包的流项在自己的流表内不存在，交换机会临时向控制器询问处理方法。被动方式是在交换机收到包的时候，再向流表里添加流，所以当只使用一个控制器集中管理的时候，域中所有的交换机都向管理这个域的控制器进行查询的话，控制器的负载会变高。

控制器的负载

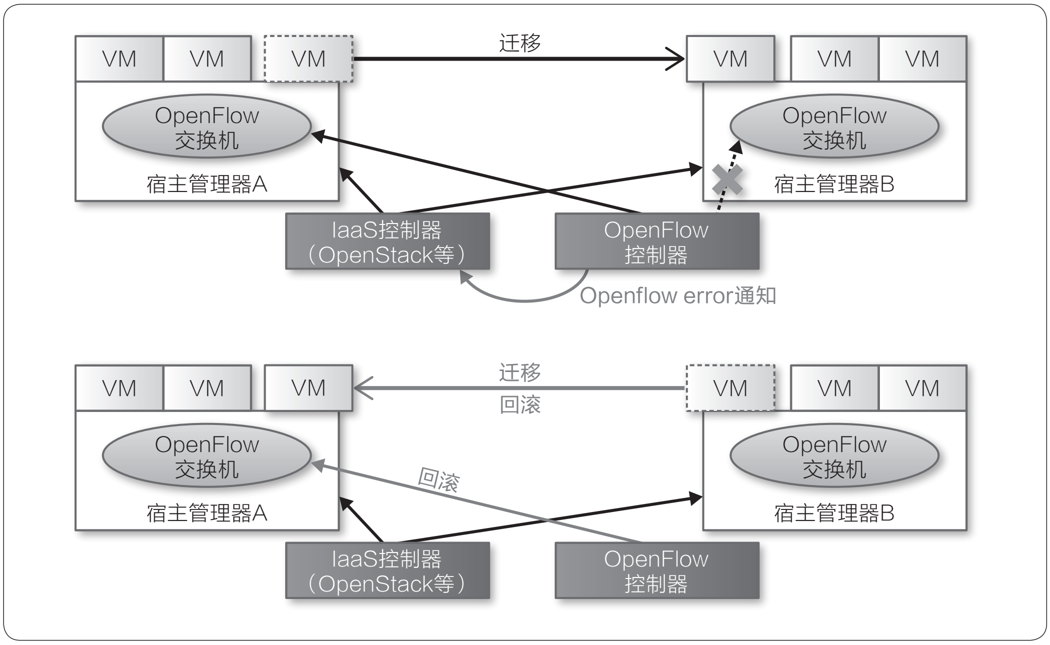
就算是主动的情况下，预先把所有的流处理内容都存储下来也是不现实的。当接收到流表中不存在的包的情况下，就需要向控制器询问包的处理方式。虽然与完全的被动方式相比，向控制器的查询数量会更少，但当事先制定的流控制项很少的情况下，就会增加向控制器查询的次数。因此，在实际运用中，控制的流项较少的情况下，应将其他包按照惯例使用协议分层转发处理作为默认行为，或者使用通配符制定某些条件，并尽量扩展该通配符的条件。这样，仅在控制器端添加流，针对某些流的处理要进行更新的时候，才会产生交换机和控制器之间的通信，从而控制器能在负载很低的状态下运行。

流的处理机制

这样的话，就留下一个问题，即在一个控制器管理所有交换机的情况下，为了实现即使修改流，交换机也能正常地基于流把包传递下去，通信路径上的所有交换机，对于该流的处理都需要同步修改。此时，若路径中的某一个交换机和控制器的通信被中断的话，针对此流的包的转发就会产生冲突。在OpenFlow中，会定时使用存活检测确认所有相关联交换机间的通信是否顺畅。但是（根据实际的实现情况）不能保证在流更新前一定会执行存活检测，在活检测的间隙若执行了流更新，导致出现了没有正常执行流更新的交换机的话，就必须回滚所有的流更新。

这个阶段回滚流更新不会产生大问题。这是因为，和流相关的处理在交换机和控制器之间已经关闭了。但是，针对IaaS这样的云平台使用OpenFlow的情况，会通过触发虚拟机实例的启动、停止、移动来执行流的变更。因此，网络这边的流变更会失败，而OpenFlow这边的回滚处理最终会通知到IaaS控制器，在IaaS端的虚拟机也需要将实例操作进行回滚。这样的话，

▼图3　迁移回滚



包括IaaS平台在内，若要维持网络拓扑无冲突的完整性，维护目标可能会变为负担相当大的网络（图3）。

流驱动网络的问题

从网络协议的视点

对于OpenFlow来说，另一个致命的缺陷不是在协议层执行转发行为，而在根据流运行的部分。一开始说过，OpenFlow是和TCP/IP完全不同的协议，它把TCP/IP下面四层定义为一个流层。虽然流层是一个独立的协议层，但实际是由TCP/IP下面4层组成的。换句话说，虽然转发是通过交换机端口、帧的MAC地址和VLAN-ID，IP地址或者MPLS标签，还有TCP/UDP端口等多个标识符的组合来进行判断的，但这里也会出现问题。

一般网络的性能是以交换机、路由器、负载均衡器、服务器等各层级为单位独自调优，整体性能因此而获得提升。当然，并不是所有层级都在同一层面上进行优化，有时，这会引出服务器的CPU和内存的处理能力超过网卡的理论传输速度的数据I/O的情况，有时，交换机端会支持服务器CPU和网卡不能带来的广带域端口。

OpenFlow必须不区分层级，在流层全体进行调优。因为TCP/IP可以以各个层级为单位进行调优，可以独立把各层级的性能提高，只要保持各个层级之间的接口，就可以同时并行开发。但是，流层必须把4个层级的处理都实现在一个接口上。因此，网卡、交换机、路由器、服务器的OS这些层级就不能独立进行产品开发和

评估检验了。由于流层的开发和调优都由单个开发团体来进行，和分层级的开发相比，可能有一点不方便。

现阶段还不能得出在网络协议上，垂直分割型和水平分割型哪种在开发工程和性能提高上更有利的结论，但确定的是，开发工程是完全不同的。

运维方面高负载的可能性

另一个是在运维方面的缺陷。OpenFlow使用流层来处理包转发，组成这个流的字段（MAC地址、IP地址、VLAN-ID等）多种多样，根据任意字段的组合定义出来的流数量也变得相当巨大。接收到的包从巨大的表格中检索出相匹配的流，会产生很大的负载，也会一定程度上影响转发的效率。Open vSwitch是一个OpenFlow的软交换机。它把流项哈希化，以此来降低表格的检索负载。

并且，在OpenFlow交换机标准1.3版本中，把流表进行，拆分，并将其管道化，从而使得流匹配处理性能得到提高并促进了动作集（actionset）的多样化（图4）。

但是，Open vSwitch把流项哈希化必须保持与包的匹配完全一致。因此，需要在控制器端的程序，指定IP地址和物理端口、理论端口的范围，再将其逐一拆分开，在流表中记录各自的流项，这样，整个表就会变得相当庞大。如前所述，随着项数量的增加，会增加表格检索、匹配处理的负载，同样也会影响到转发的性能。

另外，流表的管道化上也有一定的问题，即使没有流表中对应的项，也必须把接收到的包和所有的流表进行匹配。这样虽然可以缩小表的大小，但导致了表项匹配处理的增加。

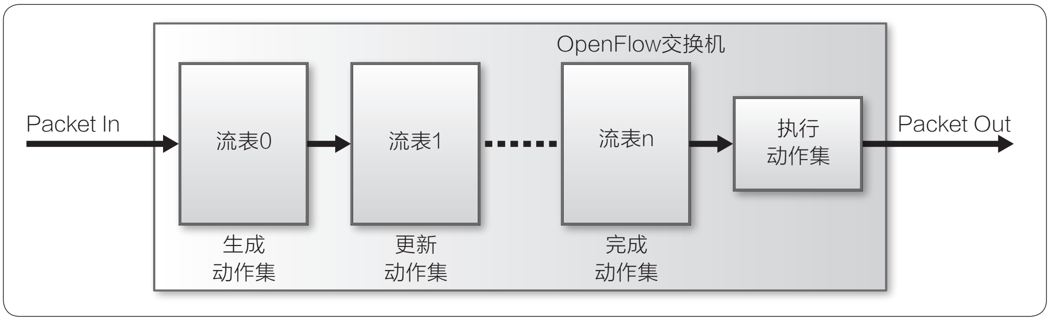
为了降低流项匹配的负荷，最小程度的包进行流处理，将其他的包进行默认转发也是一种手段，但这样的话，就和使用流进行包驱动的OpenFlow基本概念有些偏离。最终，若网络都用流进行驱动，会严重影响网络性能，为了维持网络的性能，就会产生尽量不使用流驱动这样的矛盾了。

可编程网络的问题

意外情况下无法应对

OpenFlow的其中一个缺陷是对于包的处

▼图4　流表管道



理使用的是可编程方式。网络通信隧道的建立和对包的处理使用if-else条件判断定义，这样的方法本身是没有问题的。这应该是一个受欢迎的功能架构。问题是，因为是可编程的，要让网络运转就必须要编程这点。为了让网络整体通过OpenFlow进行流驱动，必须对所有的OpenFlow交换机、针对所有的包、对所有的流处理进行编程。网络整体的流定义数量可能因为交换机数量和包的数量的乘积，变为一个天文数字。比较现实的是，不对所有的交换机和包，只对部分的包以及某个具体区域的交换机进行流定义。但这样做又会出现为了活用流，而增加了程序定义工作，或为了减少运用负载而减少程序定义等偏离流驱动网络初心的矛盾。

并且，因故障或误操作等原因，不能正常传输包的时候，为了找到故障原因，不仅需要检查各个交换机的状态和交换机之间的状态，还需要检查控制器端的程序代码。实际使用中，名为NOX和Trema的控制器和框架上，并没有这样复杂的编程代码，即便如此也需要对各模块、类、方法进行分别调试。

难以探寻问题原因

一般构建网络的流程是，设计拓扑结构，在价格和性能方面考察交换机和路由器并进行选型，搭建后配置各个交换机，依次进行连通性检查，最后投入实际使用。有问题的时候，在对个别机器的状态和相邻系统间的联通进行调查后，修改定位到的机器的设置或一组机器间的设置来进行修正。

但是，使用OpenFlow构成的网络，从搭建交换机，逐一设置端口的IP地址和默认转发设置，接着针对这些交换机进行流定义都需要编程来完成。针对故障当然要检查个别机器，但在交换机上不可能有所有的流项，未被匹配的流会被定期从表中删除，或者没有接收到流控制对象的包时，可能也不能从控制器获得那个流项。因此，针对流全体的检查，不仅包括各个交换机的物理状态和基本设置，流表的检查，还需要检查控制器端的流定义，还有程序代码这些东西。处理的工作量也随之增加。自然而然，为了找出故障原因所需要的步骤也增加了，因此修复故障所需的时间也可能大幅增加。

综上所述，不是分层级的包转发，而是通过流表依赖编程来控制的系统也有很大的潜在缺陷。

高效使用OpenFlow的方法

那么OpenFlow是失败的范例吗？笔者并不这么认为，相反，可以利用前面指出的致命缺陷，构想出一个非常高效的利用方法。

从缺点到优点的转换

总结来说，OpenFlow的致命缺点如下：

1. 控制和数据层分离

2. 把4个层级当作一个层级进行转发

3. 可编程化

换句话说，只要避免这些缺点，发挥OpenFlow的特有功能就好了。因此：

1. 不分离控制和数据层

2. 不使用流进行转发

3. 不编程

以这样的前提来活用OpenFlow的功能。OpenFlow的功能里需要特意提一下的是，不使用流进行转发，而是使用流进行包重写。或者说，接收到匹配某些特殊流的包时，可以重写这个包的任何头部字段，还能添加删除字段。或者生成新的包，并将其转发或返回。

还有，“不编程”这个条目，是不以编程来完成流转发的意思，把编程限制在字段重写，或者包生成的话，就不会对使用造成太大的负担了。

针对上面1~3的条件，再加入包字段的重写处理的话，自然而然能展现出这种处理的场景了。现在的网络域都是在域内统一使用相同的规则来运行的，即AS、IP地址块、VLAN、叠加等等。在类似于云系统等连接多个域而成的单个虚拟计算系统的情况，各个域使用不同的规则会出现问题。

例如，要让使用不同的VLAN-ID分配规则的域之间无缝连接，就需要VIT。未被虚拟化的物理网络和使用前一回解说的VXLAN和NVGRE虚拟化的叠加网络之间的通信，需要匹配物理网络端的VLAN-ID和VXLAN的VNI、NVGRe的VSID。被分配相同IP私网地址的跨域通信，则需要能把各自地址映射为虚拟IP地址的NAT这样的功能（图5）。

活用OpenFlow

作为使用多个规则构建的网络域之间通信的网关，OpenFlow交换机是非常有效的架构。基本上异构域之间通信的网关，每个域使用一台或几台就足够了。因此，数据层的交换机和控制层的控制器可以一对一来使用，还可以在一台物理机上或服务器上共存。这样，在实例上看是各自存在的，但是从维护者来看，和以前的独立型网络设备是一样的。因此，不需要关注交换机和控制器的连接性以及控制器端的交换机的区别，也就避免了层分离的缺点。

再者，网关只需要根据流匹配包，并对头部进行必要修改即可，不需要对网络整体的所有的包都进行流驱动，只把一部分的域间流量作为处理对象，还可以把对转发性能的影响控制在最小程度。

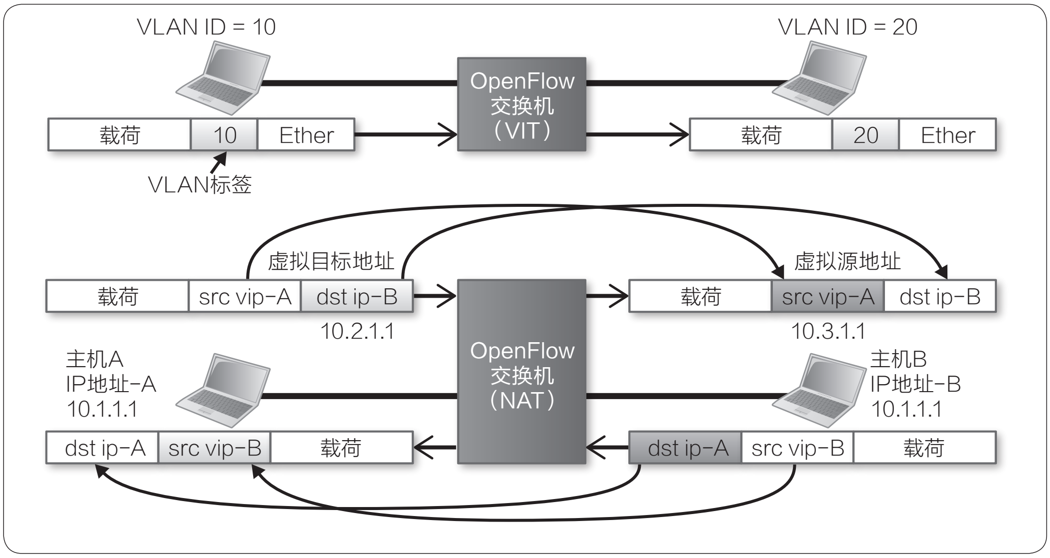
和VLAN网的兼容性

不需要对包进行条件判断和复杂处理，只用编程处理匹配流的定义和头部字段的更新，这样由编程所带来的开发工时和查找故障原因也不是大问题了。例如，在非虚拟网络和VXLAN网络之间的联通上，需要重写包应用数据单元上VXLAN头的VNI。虽然OpenFlow交换机不支持对应用数据的重写，但只要OpenFlow交换机的数据路径中支持VXLAN的话，在VXLAN的实例中就会设置了这个VNI。OpenFlow交换机检查包的VLAN-ID，并将包转发到负责传输该VLAN-ID至对应VNI的端口，就能联通非虚拟网络与VXLAN域。因为基本上OpenFlow交换机是可编程的。

消除数据中心的烦恼

现在数据中心和云平台遇到的最主要的问题不是在包转发上的自由度上，而是如何在无缝地、非大幅度变更工作流和框架，且不增加故障对应风险的情况下，在异构域之间进行连接通信。对这个问题，可以说OpenFlow是最合适使用的技术、协议。

▼图5　作为域间网关使用



最　后

最初笔者构思本连载的时候，仅想到了OpenFlow的协议和范例，并没有特别考虑其致命缺陷。但是，在请教各方朋友的意见后，调查分析了文献、论文、规范、代码，笔者意识到作为OpenFlow优点被宣传的那些特点，同时也全都是它的致命缺陷，OpenFlow有着在架构上的本质问题。意识到这些后，就自然而然地想出了高效使用OpenFlow的方法。

笔者曾经在名为OCDET（Open Cloud验证试验task force[②]）的社区互动中实现过在异构云平台间连接的隧道网关，现在笔者意识到VIT隧道已经在OpenFlow交换机上实现了。因此，得出了OpenFlow不是组成LAN整体的技术，而是可以高效地在异构域间无缝连接的技术这个结论。针对各种技术、协议、实现，不是只从优点或缺点单方面来判断，而是在优点里关注缺点、缺点里关注优点这样来分析，就能活用各种技术了。

② http://www.ocdet.org/

文/Debian开发者：山根秀树（henrich@debian.org） 译/芳龙

Debian 7.0 桌面周边的变化

Debian热点

Debian 7.0发布！

继续上期的连载，我们来介绍Debian 7.0 Wheezy的特征。这一次主要讲桌面周边的变化。

桌面的主题是joy

Squeeze的桌面是彰显流行的spacefun，而Wheezy的摇身一变，成了雅致的黑白色的joy。

如果升级成Wheezy的话，主题图片会自动切换成joy。但也许有人对以前的spacefun还很怀念。包含这些图片的desktop-base版本解决了这个问题，它带有spacefun主题的图片。想使用以前的spacefun图片的人，如果是GNOME 3的话，就选择系统设置的“背景”，点击窗体左下的“+”，直接指定以下文件即可：/usr/share/images/desktop-base下的spacefun-wallpaper.svg或者spacefun-wallpaper-widescreen.svg。

标准桌面是GNOME 3

标准的桌面环境是GNOME 3.4，除此之外还有KDE 4.8.4、Xfce 4.8和LXDE 0.5.0。过去有传言说Xfce才是标准桌面，但实际上只是因为当时开发者没有把GNOME收录在第一张CD中，而是试着放在了Git上而已。其后，作为Debian版本存档管理团队ftpmaster的一员，Ansgar Burchardt通过把包压缩形式改为xz从而减小了容量，GNOME又一次作为标准桌面出现了①。但是，GNOME 3.x（它的GNOME Shell）与GNOME 2.x相比外观有很大的区别，因此对

① 虽说标准桌面是GNOME，但安装时其实可以指定其他桌面环境。

▼图1　Squeeze的桌面（spacefun）



▼图2　Wheezy的桌面（joy）



此有抵触心理的人也不在少数。此时，可有如下的对应策略。

① 不管怎样先适应[②]……

② 使用GNOME 3的classic模式[③]，操作感与GNOME 2相近。

③ 变成与Xfce4等的6.0（Squeeze）一样的桌面环境。

不管标准桌面环境是什么，可以自由选择桌面环境是Debian比较好的地方。有时使用的3D图形芯片加速功能不起作用时，就会出现如图3[④]所示的信息，系统会强制选择classic模式[⑤]。这种情况在虚拟机中尤其常见。

② 因为桌面操作大部分都是习惯性的，刚开始时如果有些操作实现不了就会产生不满。不过，渐渐习惯后，GNOME 3还是很好用的。当然，GNOME的变化还是有些太激进了。

③ 不久前被称作fallback模式，登录时在窗口左下的选择项里可进行选择。只要选择这个，就会出现与GNOME 3的GNOME Shell的操作感不同的、与过去的桌面相近的菜单。

④ 右边没有完全表示出来是因为这是测试用的虚拟环境，所以弹出框没有显示在正确的位置上。

⑤ 不过，最新的GNOME 3在图形芯片3D加速功能不能用时，通过利用llvmpipe进行软件处理可使GNOME Shell不管在什么环境下都可以启动。但是，由于llvmpipe的处理太繁重，因硬件、视频驱动而不支持3D加速功能的环境，强烈推荐使用classic模式。

▼图3　3D加速功能不能使用的情况



向LibreOffice转移

Office组件OpenOffice.org被替换成了LibreOffice 3.5。OpenOffice.org版本和LibreOffice版本的维护人员Rene Engelhard在The Document Foundation设立之初就是项目成员[⑥]，而The Document Foundation正是LibreOffice的前身。因此这种更迭可以说是理所当然的。另外，KDE的Office组件KOffice也伴随着upstream的改名而改成了Calligra。

浏览器

主要的浏览器Iceweasel（不使用Firefox商标的版本）采用版本10，但事实上真的看起来很旧也很逊色。版本10是延长支持版，它面向法人的支持周期大约是1年，小规模的升级周期是6周左右[⑦]。但是作为上游的Mozilla方已经不支持版本10了，对延长支持版的支持已经到了版本17。关于这个问题笔者在发布之前就提出来了，结果还是没有得到解决。

之后，在进行安全性升级时，更新成了17.0的ESR[⑧]。对于像Firefox这样庞大的软件，与其继续对版本10.0做backPort（将一个软件的补丁应用到比此补丁所对应的版本更老的版本的行为），不如基于版本17.0再发布一次更好[⑨]。但是，关于扩展包的绑定方式，至今还没得出

⑥ URL：https://www.documentfoundation.org/foundation/members/

⑦ URL：http://www.firefox.com.cn/download/。也叫ESR（Extended Support Release）

⑧ URL：http://www.debian.org/security/2013/dsa-2699

⑨ URL：http://lists.debian.org/debian-devel/2013/05/msg01572.html

什么结论。而且，在想使用新版本时，可以通过追加由Debian Mozilla team提供的mozilla.debian.net的发布源来实现，因为这里随时都在更新，而且还可以提供Beta/Aurora版，所以推荐给喜欢使用新产品的人。追加发布源会在网站上产生apt line，可以将它粘贴在/etc/apt/sources.list上使用。

另一方面，Chromium（Google Chrome的开源版）现在已经发布了版本27，Windows系统使用Chrome的用户也可以安装它。

支持多媒体

过去，Debian中使用mp3编码的lame、mplayer等软件由于编解码器的专利和许可证问题，没有收录到包中。但是最近Wheezy可以使用各种各样的编解码器了。读取音乐CD的软件Sound-Juicer可以选择ogg/mp3/mp4/FLAC作为标准的输出文件格式（虽说标准是ogg，如果便携式音乐播放器不支持的话，也可以转换成mp3格式。对音质要求高的人请选择FLAC）。至于动画也因为有x264包可以实现H.264/MPEG-4的播放。在这之前，这些都是由一个叫deb-multimedia.org的外部网站来承担的[10]。今后，如果不是不得已的话，都可以由Debian的正式包来完成。

⑩ 以前是叫debian-multimedia.org，但是这个名称看上去像是Debian的官网，用户习惯性地将从此网站的发布源下载的包的错误报告给Debian，因此不得已才改名。域名debian-multimedia.org已经给了第三方，注意请不要使用。

7.1发布

修改了7.0的问题并巩固了安全性后，2013年6月15日Debian 7.1以点发布（point release）的形式被更新到网站上了。到Squeeze为止Debiand都是以第2位小数点后的数字表示点发布的，这次不知为何做出了这样的调整。关于点发布可以事先在proposed-updates上确认情况，所以运用系统的人把验证环境准备好，根据情况做好反馈就可以了。

下次发布Jessie

Wheezy发布了就意味着恢复升级了。在冻结期间，升级到unstable原则上是禁止的，取而代之的是向experimental的升级[11]。Experimental和stable/testing/unstable是完全分离的，如下所示，安装时不明确指定experimental的话，包的导入是不会成功的。

```
apt-get Install -t experimental &lt;package&gt;
```

像unstable→testing这样自动的包转换是不能执行的隔离空间的，因此使用了experimental（试

⑪ 在发布管理上，testing发生问题时，在unstable上进行修改，把冻结状态设为unblock就可以取得unstable包。但是如果安装了unstable的新版本，testing和unstable的差别就会过大，会给发布者带来麻烦。因此，向unstable的升级原则上是禁止的。但是，并没有防止向unstable的升级的设置，关于这些问题的决定权都交给了需要升级的开发者。另外，即使testing包发生了问题，如果感觉上传到unstable的包升级很困难，则采用以下方法来解决：上传加了最低限度的批处理testingproposed-updates的版本，然后通过点发布来修改问题。

验）这样的名字。

解禁了unstable的升级后，包的维护人员把向experimental升级的包接连改成了unstable。不过在较长一段时间内，unstable会反复出现比较大的变化。

在宇宙和云领域活跃的通用OS

媒体报道了ISS（国际空间站）的便携式电脑的OS由Windows换成了Debian 6.0的新闻[⑫]。这并不是因为国际空间站的工作人员喜欢Squeeze主题的spacefun壁纸，而是因为吸取了Windows蠕虫的教训。

在日本也有人关注着这个话题其中也有对Debian持怀疑态度的人。但是NASA的系统负责人（负责ISS系统的企业United Space Alliance的管理人员）称：

We migrated key functions from Windows to Linux because we needed an operating system that was stable and reliable—one that would give us in-house control. So if we needed to patch, adjust, or adapt, we could.

（从Windows到Linux的转换是因为我们需要一个稳定的可信赖的OS——其中的要素之一就是我们自己可控。我们需要打补丁的时候就可以打补丁，需要调整的时候就可以调整，需要更改的时候就可以更改。）

可见他们更需要的是对于OS的可控性，像Debian这样的OS就非常适合他们。

而且，在Debian 7发布的同一周，Google提供的云计算机服务Google Compute Engine选择了Debian 6和7作为主要OS[⑬]。由此，Amazon Web Services、Windows Azure等有名的云服务也可以立即使用Debian或提供相关支持。

⑫ URL：http://www.zdnet.com/to-the-space-station-andbeyond-with-linux-7000014958/

⑬ URL：http://googleappengine.blogspot.fr/2013/05/bringing-debian-to-google-compute-engine_9.html



我家宝贝儿综合症：关于用户咨询的思考

大村 芳树
Yoshiki OOMURA
红帽（股份）全球支持服务
高级技术支持工程师
译/王凤波

自我介绍

大家好！我是红帽全球支持服务部（Global Support Services，GSS）的大村。初次见面，请多关照。根据我所在的部门名称大家就可以知道，这是一个产品支持部门，专门负责Red Hat Enterprise Linux（RHEL）的技术支持工作。全心全意为用户提供技术支持服务是我们的使命（这是开场白）。

本专栏刊载过多名红帽公司的技术顾问、SA等知名资深技术专家撰写的文章。这一次，红帽与技术评论社开诚布公，广开言路，决定给资深技术专家之外的普通人士一次执笔的机会，而这支白羽之箭幸运地射在了我的身上。

但是我后来才发觉，也许是由于业务种类上存在着这样那样的限制，本想为大家介绍一下具体的工作内容，提起笔来却发现实在是爱好由来下笔难啊。因此，这一次姑且先为大家介绍一下在惠比寿的某个地方经常为人们所悄声谈论的"我家宝贝儿综合症"。但首先要向大家声明一下，仁者见仁，智者见智，不同的人可能会有不同的看法，所以这里所讲述的内容纯属一家之言，如有不妥之处，还请多多包涵。

谈论"我家宝贝儿"的前提条件

想必本书的读者们都有着不同的技术背景，对于系统而言每个人的参与方式也不尽相同，但是其中肯定有很多人曾经有过这样的经历，那就是受理来自客户或者用户的提问并作出相应的解答。笔者从事Linux相关工作已有15年之久（说实话，具体是从哪份工作开始的还真记不清了）。入职红帽公司也已经是第7个年头了。最初任职于培训服务部门，负责培训工作，后来调任到技术支持服务部门，一直工作到现在。笔者在加入红帽公司之前就曾经做过咨询、培训、产品支持等方面的工作，工作形式虽然不尽相同，但是那段时间，每天都在直接接触用户，受理来自用户的咨询。现在回想起来，我发现Linux的用户层好像发生了相当大的变化，这一点感触颇深，也许是因为用户层的拓展本身就是一件令人可喜可贺的事情吧。

"我家宝贝儿"的父母是谁？

这样说可能要受到资深技术专家们的批评了，15年前我作为用户，曾经将Linux想象成宠物。当它不听话的时候，我就按照自己的方式自娱自乐地去哄它："嗯，是否完全正确暂且不管，你好像终于消气了。"有时候也会因为处理不好而故意惹它不高兴。还有就是与其他人之间共享宠物的信息。饲养同一种宠物的父母们（？）

之间会做这样的交流："我家宝贝儿最近正处于反抗期呢"，"你给它正确喂食了吗？在我们家呀……"通过这样的沟通进一步加深了彼此之间的认识。毫无疑问，"我家宝贝儿"的父母是用户，也许正是因为首先有了这个前提，我才一直有意无意地把"我家宝贝儿"即Linux想象成宠物吧。

随着时间的推移，红帽公司的产品（当时笔者还是个局外人）名称中也开始加上了Enterprise这一中间名字，逐渐地迎来了一股红帽宠物热。而真正感受到Linux系统用户数和用户阶层的增加所带来的变化，则是在笔者进入红帽公司之后的事情。

"我家宝贝儿做了这样的事"案例

言归正传，接下来我们将话题返回到"我家宝贝儿综合症"之上。我们经常可以看到这样的支持案例。这只不过是日常发生的一个咨询案例，具体的log日志（实际咨询过程中受理的咨询对象等）这里就忍痛割爱，不为大家作具体介绍了。喜欢看日志的读者朋友，还麻烦请参考一下自己宝贝儿中的日志。

> Q-1:
> 系统报出了如下消息。请问有什么影响？应该采取什么样的应对策略？
> <……1行左右的日志消息引用>

同样的咨询可能存在多个不同的解决方法，在这个案例中我们不妨把解决的最终目标放在"影响"和"应对策略"之上。正如开场白中所讲述的那样，GSS的使命是全心全意为产品用户提供周到的技术支持服务（"支持服务的范围"等这些生硬的话题，我们就略而不谈了）。但话说回来，这只不过是一个简单的日志引用而已，却需要站在"影响"和"应对策略"的高度上来予以解答，实在是有些勉为其难。

假设我们根据该日志所提供的相关信息，在尽可能的范围内作出这样的回答："该消息只不过是info级别的日志，如果XXX的运行上没有什么问题的话，通常不必在意。"（为了便于说明，这里我们假设该日志的级别为info或者debug。）

但是，有的用户可能会像下面这样进一步追问。

> Q-2:
> 为什么说通常没有什么影响，请进一步说明。既然没有影响，为什么还要写日志呢？

想要让此次咨询最终画上圆满的句号，有一个无论如何无法回避的难题，那就是我们必须得向客户说明这样一个问题："无论是info级别的日志还是debug级别的日志，在动作的追踪上都是有用的，但是却无法根据该日志掌握系统的状态。"（嗯……不知道大家是否能理解？）

对"我家宝贝儿做了这样的事"的解读

对于上述Q1和Q2，我们也可以通过这样一个比喻来进行解读。假设一天早餐，有一个小孩在日记中这样写道：

> 昨天晚上吃了海胆盖饭，两大碗

关于父母是否应该看孩子的日记，我们姑且不作讨论，这个孩子的父母在看了日记之后，没有带上孩子，单独找到了医生，向医生讲述了日记的内容并询问："我家的宝贝儿是怎么了？这可怎么办啊？"对此，医生会询问道："你家宝

贝的情况怎么样？”经过一番询问之后，医生也只能这样回答：“原来是吃了海胆盖饭啊，我想如果孩子身体没有什么异常的话，就不用在意。”

对此，孩子的父母却进一步追问道：“为什么身体没有什么异常就不用在意呢？”当故事情节发展到这里，父母的兴趣显然已经不在孩子的身体之上了，而是已经转移到了对于日记的解释之上。

如果日记中写了“肚子疼”，那也就无可厚非了。真正的问题应该不在于日志中所写的内容，也不在于父母偷看了孩子的日记，而应该在于孩子的身体状况。

与上述日记的例子不同的是，查看自己管理的服务器的日志是一种必要的行为。如果平时不查看日志，就无法根据日志了解某些征兆，所以通过查看日志来监护“我家宝贝儿（被管理的服务器）”的状态是非常必要的。也许通过这一工作可以读出类似于“肚子有点不舒服……”的消息，发现潜在问题的征兆。对于孩子而言，要察其言观其色，观察其早安问候时的精神状态，并且还要在此基础之上留意他所写的日记或者他所说的话语。这些都是必要的，而且每一项都不可或缺。同样，对于被管理的服务器而言，养成留心查看日志的习惯也是非常重要的。

“我家宝贝儿，您家宝贝儿”

是否一定要举海胆盖饭的例子暂且不谈（有时也会列举咖喱饭的例子），事实上笔者在很多实际咨询案例中都曾经向用户做出过类似的比喻说明。另外，上面我们列举了一个关于日志输出的咨询案例，实际上我感觉有类似倾向的咨询案例正在不断增加。这里所说的倾向是指将被管理的服务器完全看待成“您家宝贝儿（红帽的宝贝儿，或者其他商家的宝贝儿）”的一种倾向。

这未必是一种错误。确实，根据不同的见解，即使我们生产的产品已经转交他人之手，归他人所有，也未必不可以说“这些组件或者已经发布的产品就是我家（红帽）的宝贝儿”。

也正是因为如此，如果有父母说“我家宝贝肚子疼”，我们就会建议他“请您给他服用这副药试试”，竭尽全力为父母们提供周到的服务，履行我们红帽支持服务部门的使命。

不过，对于“离开我家（即红帽）到了别人（即用户）家的我家宝贝儿”，我们却不能直接去跟他们见面，当面关切地询问“怎么了？哪里不舒服？”也不能每天都留心观察他们的脸色和聆听他们的话语。

重新强调一下笔者的愚见，父母对于“来自红帽公司的宝贝儿”进行培养和监护，而我们红帽公司所担当的角色则相当于对父母进行监护的祖父母（虽然这个家谱看起来有些奇怪），或者相当于宠物商店的店员或兽医，我们的职责就是对子女培养提供支援服务。

为此，我们红帽向父母们提供各种各样的支持服务。我们还面向父母们开办如何养育子女的培训课程，有时对父母们甚至会采取一种近似说教的语气，比如说“我们红帽家的孩子吧……”等。负责检查孩子受伤情况，以及帮助不知如何培养子女的父母们，这些就是我们所提供的全球支持服务。另外，如果有更深层次的要求，例如“希望使我家宝贝儿变得更加漂亮”，“希望把我家宝贝儿培养得更加优秀”等，还可以向我们的资深技术专家们（相当于宠物界的资深育种专家）进行咨询。



第17回

Linux 3.10新功能——减少时钟中断的NoHZ

文/青田直大（AOTA Naohiro） 译/kaku

面向NoHZ

CPU为实现多任务，其内核时钟会中断，比如每1000分钟中断1秒，以便于对任务进行预定处理。但是，如今一般都使用多核处理器，所以也可以设定一个CPU只执行一个进程。这种情况下，若没有其他可以活动的进程，就可以认为CPU的时钟中断是没什么太大作用的。所以，以完全消除CPU时钟中断为目标的NoHZ功能的开发一直在进行着。在Linux 3.10的RC版本中，作为NoHZ迈出的第一步，它已整合至patch中。关于NoHz，本回我们来了解一下。

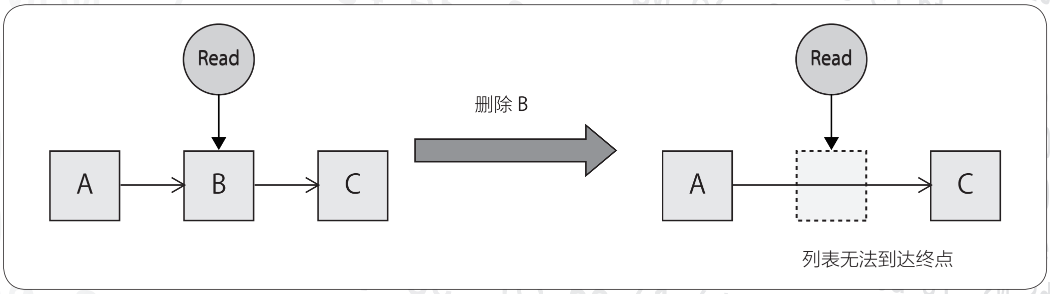
RCU

因为RCU的回调和NoHZ也可关联起来，所以在解说NoHZ之前，先来了解一下RCU。

RCU是Read-Copy-Update的缩写，是Linux内核中作为Readers-Writer lock的替代品而常被使用的同期工具。让我们通过比较它和Readers-Writer lock的差异，来看看RCU到底是什么。

Readers-Writer lock是为读写共同数据而用的锁。按图1所示的列表来思考一下吧。某个线程在读取列表时，若其他线程删除列表中的元素，此列表就不能正确到达终点。这就需要用Readers-Writer lock来限制列表的读写操作。这

▼图1



个锁在读取端并不改变列表，只有读操作，所以不论有多少都可以同时进行。

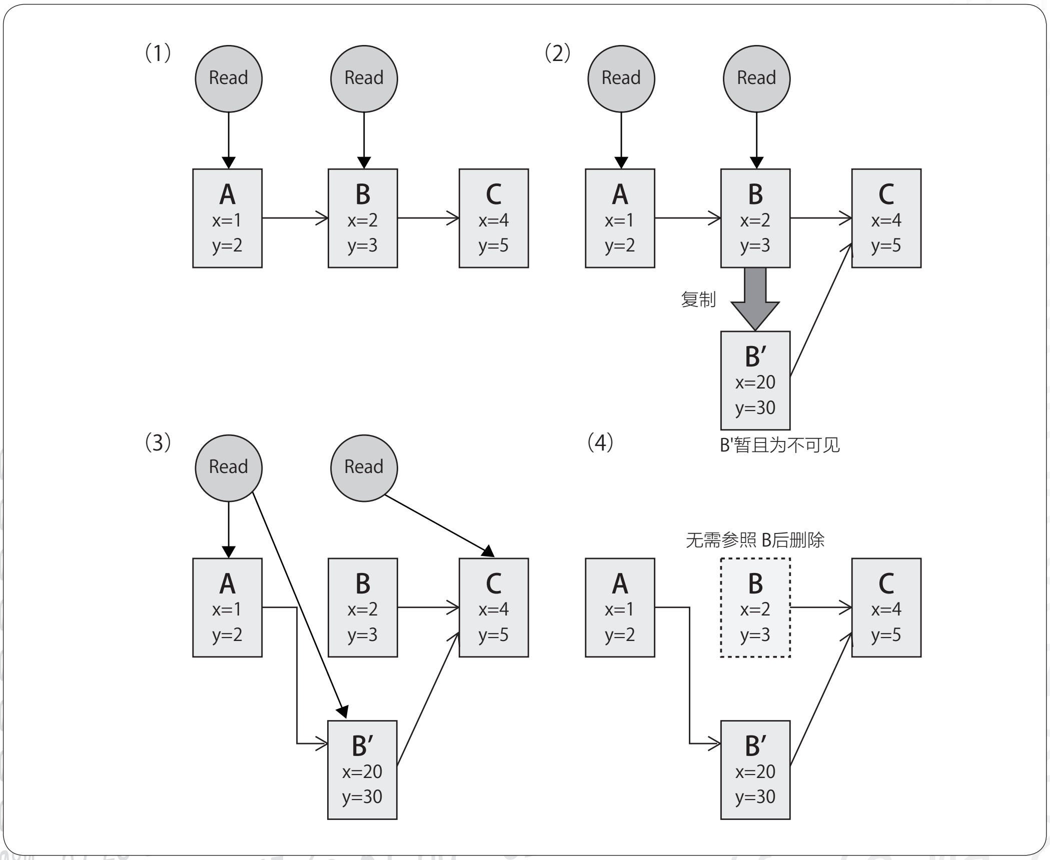
另一方面，正在进行写操作时，若其他写操作/读操作同时存在，就会出现刚才提到的问题，所以只允许存在一个写操作，并且不可以与读操作同时执行。需要注意，Readers-Writer lock时的读操作和写操作不能同时执行。也就是说，读操作端有大量常驻进程时，一旦出现写操作，读操作的动作就会完全停止。

对于RCU来说，其明显特征就是读写操作可以同时执行。它是如何解决刚才的读写问题的呢？如图2所示，我们来介绍一下如何更新列表中的元素。初始的列表状态为（1），这个列表的所有元素都可以被读操作看到。（2）将B复制成B'，并更新B'的数据，此时，读操作是看不到新数据的。接下来（3）将从A到B的指针切换到B'。因指针更新是原子操作，所以此操作可以确保无误。那么，这里B自身因还未删除，所以参照B的读操作，还可以继续正常访问。（4）等待读操作结束时，删除原来的B。这时读操作和写操作就可以同时进行访问了。

这样，RCU中的替换就可以简单实现了。但还存在一个大问题：如何判断查看原来数据的读操作是否存在。为解决这个问题，Linux内核加入了如下两个部分。

首先，读操作的读取部分（RCU参照的

▼图2



临界区域）由函数rcu_read_lock()和rcu_read_unlock()提供保护。在RCU参照的临界区域中，禁止抢占式多任务处理，即在这中间不发生上下文切换。接下来准备一个和CPU个数相同的bit数组，用synchronize_rcu()开始Grace Period（等待Grace Period开始时存在的读操作结束的时间），见图3。

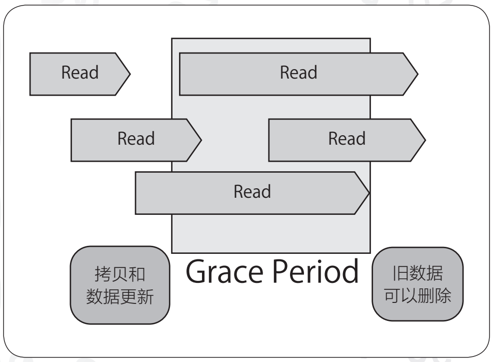
这个函数会准备与所有CPU对应的bit数组的bit。而且在上下文切换时，与CPU相当的bit会被全部清空。这样就可以知道所有CPU在何时执行上下文切换了[①]。

在图3中写操作使用synchronize_rcu()等待读操作结束后进行删除处理。与此相反，若先登录执行删除处理的回调函数call_rcu()，之后其他线程等待读操作结束后执行删除处理也是可行的。这样将Grace Period处理移给其他线程的话，写操作就不需要等待，从而改善，延迟时间。Linux 3.6以后大部分的Grace Period都移动到了内核线程。

而且Linux 3.8开始加入了callback-free CPU

① 这个bit数组的使用方法在以前的应用中，由一个bit数组锁来防止所有CPU发生竞争，如今变为了Tree RCU的测量方法。

▼图3



的patch。若开启这项功能的话，就可以在boot参数rcu_nocb=中指定“不执行”RCU回调的CPU。本来应该此CPU执行的回调，会由其他CPU中活动的rcuox/N（x是RCU的种类，另外还有b、p、s等，N为CPU号码）内核线程负责。这样既可以限制执行这个内线线程的CPU，还可以改变优先度。使用callback-free CPU功能，使某个CPU内核工作负荷最小化，这是NoHZ很重要的一个特点。

总结一下到此为止有关RCU的话题。

- RCU是一种同期工具
  读操作锁的成本很低
  可以同时执行读写操作
  但是必须要在某处执行删除处理
- 删除处理可以作为回调在后面执行
- CPU可以设置成不执行RCU回调

dyntick-idle mode

下面，开始正式介绍NoHZ吧。以下两种情况可以减少时间间隔：一是CPU什么都不做进入空闲状态时；二是所有CPU中只有一个可以执行的任务时。空闲状态时，可以减少时间间隔就被称作dyntick-idle mode。这是早在Linux 3.8之前就加入的功能，在这里也简单看一下吧。

时钟中断的最终目的，是为了CPU执行多个任务时，顺利实现切换。因此在不需要做任何事情的空闲状态时，没有必要让CPU进行时钟中断。

关于dyntick-idle的优点和缺点我们也来看一下。优点就是减少时钟中断后，CPU无需在

执行那些无用的工作。这样CPU就可以长时间进入低电力消耗的深度空闲状态（图4）。

未使用dyntick-idle减少时钟中断的内核，比减少中断的内核耗电速度要快2至3倍。还有，很多虚拟机在启动状态时也要开启dyntick-idle。没有dyntick-idle时，每1000秒一次的中断处理只在虚拟机中执行，所以即使主机的CPU处于什么都不做的空闲状态，大部分时间也被中断处理给占去了。

缺点方面是怎样的呢？dyntick-idle mode在进入空闲状态时，会检测下一个预约的时钟事件。也就是说，若事件稍稍提前的话，就会停止时钟中断，将时钟设定为此事件的执行时刻，并可以从空闲状态恢复。接下来，恢复后再重新开启定期时钟。这样，在进/出空闲状态时，就需要多余的处理。对于需要及时应答的系统，这个延迟就会成为问题所在。

adaptive-ticks mode

如前所述，只有一个可以执行的任务时，时钟中断是没有什么意义的。这时，取消时钟中断（进入adaptive-ticks mode）的CPU，被称为adaptive-ticks CPU。内核选项CONFIG_NO_HZ_FULL_ALL没有被设定时，默认就不是adaptive-ticks CPU，要在内核启动参数nohz_full中，指定adaptive-ticks CPU。相反，CONFIG_NO_HZ_FULL_ALL被设定的情况，默认CPU0以外的所有CPU都会变为adaptive-ticks CPU。若使用nohz_full，就会被覆盖。其对应关系如表1所示。

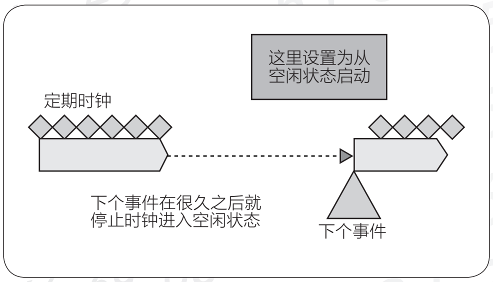
这里请注意，CONFIG_NO_HZ_FULL_ALL时的CPU0，不是adaptive-ticks CPU。这是因为，如果所有CPU的时钟中断全部停止的话，通知当前时间的函数gettimeofday()就不能返回正确的值了。这是与dyntick-idle的不同之处。dyntick-idle的话，若所有的时钟全部停止，就不存在进行中的进程，所以不会发生问题。

接下来，我们来看一下adaptive-ticks mode的缺点，简单列举如下：

- POSIX CPU时钟会变得不准确
- 进程执行时间等统计信息会变得不准确
- 定时的LA_BIAS功能会无效
- 大多perf事件不能同时获取

因运行中的CPU时钟中断全部停止，不能进行时间管理，时钟进程执行时间的统计也会变得不准确。另外，因CPU的运行时间无法获取，LA_BIAS功能也无法实现。所以，这个功能也变为无效了。

▼图4　dyntick-idle



▼表1　内核选项和nohz_full的关系

	nohz_full=空	nohz_full=有指定
没有CONFIG_NO_HZ_FULL_ALL	完全没有	只有被指定的CPU
有CONFIG_NO_HZ_FULL_ALL	CPU0以外所有	只有被指定的CPU

关于最后列出的多个perf事件，在之前的文章也曾提到过，用CPU可以监听的事件数（每个CPU的种类都是固定的）是有限的。若想监听超过界限数的事件，perf就会将登录到CPU的事件定期进行循环。当然这里“定期进行循环”需要时钟，所以adaptive-ticks mode这个功能也就不能运行了。

NoHZ和RCU

让我们来谈谈NoHZ和RCU的话题吧。RCU存在处理回调的部分，若有RCU回调时，此CPU基本上是不能进入NoHZ的。为解决这个问题有两个方法。

第一个方法，设置内核选项的CONFIG_RCU_FAST_NO_HZ，这样的话，即使拥有RCU回调的CPU，也可以进入NoHZ。但是，为执行RCU的处理，CPU要以平常时间间隔的4倍来启动（比如通常设置的时间间隔为每1000分之1秒的话，该值就会变为250分之1秒）。

另一个方法就是在RCU中解说过的callback-free CPU。使用此方法，该CPU就完全从RCU回调中解放出来了，因RCU而不能进入NoHZ的情况也就不存在了。

Linux 3.9中因导入了RCU callback-free CPU，所以NoHZ可以充分发挥它的作用。

NoHZ的未来

Linux 3.10中加入adaptive-ticks后，NoHZ的功能大为进步。但是，仍然还有很多改进的余地。其中之一就是对应32bit。这个patch好像已经有了，所以我想可能不会花太久时间。另外一个就是在adaptive-ticks mode中，把时间间隔变得更小。现在，进入adaptive-ticks mode后时间间隔变为1秒1回，与本来的1000分之1秒1回相比虽有很大的进步[②]，但还不能完全切断时钟。这需要修改内核中几个核心部分，如定时执行时间统计的部分。

perf user-space return probe

另外，我们看一下本次添加到perf的user-space中的return probe功能。之前在perf中user-space的程序中已实现了probe，但从Linux 3.10开始同时实现了return probe。这个probe可以检验一个时间，即指定函数返回调用它的函数的时间。

perf probe加“-x”使用的话，就可以指定想probe的user-land的代码（图5）。这里在kernel内的函数和相同函数名的后面加上“%return”，就可以追加user-space的probe了。使用perf record命令，就可以在10秒内记录从malloc( )返回的部分。

若执行perf report，就可以看到调用malloc()的进程记录。由此可知调用最多的是consolelog.sh，一个来自bash的gmatch_wc函数的进程（图6）。

在各个入口用“Annotate...”的话，甚至能看到该函数中的事件发生位置。report中jump源会表示箭头，还挺好玩儿的，希望大家亲自体验一下。

关于userland的probe开发还一直在进行中。如果可以很容易地看到整个系统的performance信息，还是很方便的。

② 1000分之1秒的中断处理大致需要使用CPU1%的时间。

总　结

本期我们对NoHZ进行了介绍，它可以减少运行时CPU的定期时钟和电力消耗，还可去除无用的内核处理功能。

延伸阅读



[美] Jonathan Bartlett　著
郭晴霞　译
人民邮电出版社

- 世界最优秀的计算机科学专业学生都在用的编程教材
- 语言轻松，例子实用，轻松学会用Linux汇编语言编程

▼图5　指定想探查的user-land的代码

```
# perf probe -x /lib/libc-2.17.so -F （确定可以探查的函数）
…
lsetxattr
lutimes
malloc
malloc@plt
malloc_info
mblen
…
# perf probe -x /lib/libc-2.17.so malloc%return (probe设定)
Added new event:
  probe_libc:malloc    (on 0x7ed80%return)

You can now use it in all perf tools, such as:

        perf record -e probe_libc:malloc -aR sleep 1
# perf probe -l (确认已进入probe)
  probe_libc:malloc    (on 0x000000000007ed80%return)
# perf record probe_libc:malloc -aR sleep 10 (监听10秒钟probe)
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.541 MB perf.data (~23626 samples) ]
```

▼图6　perf report的执行结果

```
Samples: 2K of event 'probe_libc:malloc', Event count (approx.): 2588
 51.85%    consolelog.sh  bash                     [.] gmatch_wc
 12.36%            uname  bash                     [.] gmatch_wc
 10.82%            uname  libc-2.17.so             [.] __strdup
  9.27%            uname  libc-2.17.so             [.] _nl_intern_locale_data
  1.55%            uname  libc-2.17.so             [.] _nl_normalize_codeset
  1.00%       notify-osd  libxcb.so.1.1.0          [.] _xcb_in_read
  0.93%  at-spi2-registr  libxcb.so.1.1.0          [.] _xcb_in_read
  0.85%    consolelog.sh  libc-2.17.so             [.] set_binding_values.part.0
  0.85%    consolelog.sh  libc-2.17.so             [.] qsort_r
  0.85%    consolelog.sh  libc-2.17.so             [.] __fopen_internal
...
```

jus快讯

Japan UNIX Society
日本UNIX用户协会 http://www.jus.or.jp/
文/法林 浩之 HOURIN Hiroyuki hourin@suplex.gr.jp
译/王凤波

2013年的LL节

本期笔者将为大家介绍一下在2013年8月24日举行的第11届LL大会（Lightweight Language Event，轻量语言大会）——LL节。

什么是LL大会

所谓LL大会就是面向程序员举办的一个集会活动。该大会诞生于2003年，最初由Perl、Ruby、PHP、Python等社区举办，后来因为受到其他语言社区以及编程爱好者们的喜爱而发展成为一项定期集会，每年举办一次，到2013年已经是第11届了。最初，LL大会的议题主要集中在比较各种程序语言的差异上。但LL的初衷是“使编程更轻松，使软件更高效”，所以后来大会的主题逐渐转变为向程序员提供有价值的信息和分享编程的乐趣。下面就为大家详细介绍一下LL大会的主要特征。

- 多家社区共同运营

LL大会的执行委员会是由多个社区的成员共同组成的。这种运营模式难免会引起一部分人的担心，因为如果某一语言社区掌握了大会的主导权，就有可能导致不公平现象的发生。但好在日本UNIX用户协会jus这一中立组织也参与了LL大会，从而在一定程度上能够确保活动的公平公正。

- 每年都制定大会标题

每年LL大会都会通过标题来展示活动的主题或关键字，从而增强大会整体的统一性。

- 预售入场券

第一届LL大会在现场向与会的约150名人员收取了入场费，结果大大拖延了签到的时间。于是从第二年起，大会就采用了预售入场券的形式，将售票工作委托给了票务公司。在IT业界各种活动的举办史上这是从未有过的先例，从此这一形式也被广泛应用到其他大会中。

- 会期仅1天，但持续时间长

LL大会也曾经连续举行过两天，但是这样人们就得把整个周末的时间都花费在大会上。但如果只举行一天，且仅在白天举行的话，又实在让人难以尽兴。因此，会期由两天改成了一天，但是举行的时间却从早晨一直持续到晚上8点左右，这样与会人员就可以有足够的时间尽情交流、畅所欲言。

- 追求看得见的快乐

LL大会的主题大多涉及时下流行的技术，演讲者也多是知名人士。不仅如此，为了让大家能够切身享受到更多的乐趣，大会还进行了很多精心的设计和安排，比如选择Live House（举办小型演唱会的会场）以及职业摔跤的环形竞技场作为大会场地，在开幕式和闭幕式上使用特有的影像效果，以及开展Lightning Talk（闪电式演讲）对抗淘汰赛等等。

- 介绍编程相关书籍

LL大会一直与IT相关出版社合作，在大会中展出并销售IT相关的书籍，并且还会把这些书籍当作奖品赠送给在大会结束时举行的投球比赛中胜出的人，这个活动也成为了LL大会的标志活动之一。

- 提供无线网络

LL大会组建了一支网络小组，在设备供应商的支持下，为与会人员提供无线网络。这是在其他集会活动中很难享受到的服务，因此获得了大家的一致好评。

- 没有资金赞助商

LL大会的举办经费全部来自于与会人员支付的入场费。企业的赞助资金很容易受到市场行情或销售业绩的影响，而入场费则不同，只要有人员参加大会，就能募集到必要的经费。所以如果想确保LL大会能够长期而稳定地举办下去，通过收入场费来募集经费的方式才比较稳妥。

LL节的看点

■ 议程安排

往年，LL大会的议程通常只有一个环节。2013年为了吸引更多不同阶层的人群参与进来，特意准备了3个不同类型的环节。

- 报告（Presentation）

这是2013年的新环节，大会将邀请十多位演讲者分别进行约20分钟的报告陈述。

- 分组讨论会（Panel Discussion）

按照往年的形式，2013年也围绕倍受关注的热点技术举办了几场讨论会。

- 培训（Tutorial）

这也是2013年的创新尝试之一，主要面向那些有意向涉足轻量语言领域的程序员，由各个语言社区负责开展几场关于轻量语言基础知识的培训。

- 闪电式演讲（Lightning Talk）

这是LL大会的惯例环节。但是与前三个环节不同的是，它将在大会的收尾阶段进行。

■ 入场券销售方法的改革

2013年，大会在入场券的销售工作上也尝试了几条新思路。

- 引进支持Social ID的售票系统

截至2012年，LL大会主要靠罗森票务系统来销售入场券。但考虑到现在的IT工程师们更习惯于通过Social ID来购票，因此2013年大会将采用支持SocialID的PeaTiX票务系统。

- 双人票

即一张票能允许两人同时入场，这也是2013年的创新之一。不仅适用于情侣和夫妻，朋友或者同事之间也都可以使用。

- T恤入场券

LL大会每年都会制作一批T恤，与会人员可以选择购买T恤和入场券的套装。2013年大会将T恤与入场券分离，把T恤本身也作为入场券进行销售，并且还提供了不同的尺码以供选择。

■ 使用平面风格的会场

LL大会往年一直使用戏剧风格的会场，但是2013年因为没有预定那种会场，所以采用了展示会中经常使用的平面风格的会场。我们也把这当作是一项新的挑战，充分发挥其可以自由布局的空间优势，进行了缜密的构思与精心的策划。

“LL祭”的含意

在构思2013年LL大会的标题时，工作人员脑海中浮现出的是，在平面风格的会场中各种创意百花齐放的情景。这种无序的浑沌感使工作人员们联想到了节日的景象，于是2013年大会的标题被定为“LL节”。

在iPhone上开发应用程序

Delphi XE4 ——桌面同步开发提升生产效率

文/Software Design编辑部　译/芳龙

2013年4月发布的Delphi XE4实现了iOS平台应用的开发。说起Delphi，其作为Windows开发工具已有很长的历史了。近年来，由于多平台支持的发展，Delphi也可以进行OS X等平台的应用程序开发。

这里我们选择iOS平台，使用Delphi，具体可以进行怎样的开发呢？我们来看看吧。

iOS应用程序开发步骤

利用模板进行开发吧

选择Delphi XE4的菜单：File–New–FireMonkey Mobile Application–Delphi。因为Delphi已经事先准备了典型的移动应用程序模板，通过新建这样的应用程序，可以简化一些最初的开发步骤（图1）。此处要选择Blank Application。

这样，iPhone的图形化设计画面就显示出来了（图2）。除了iPhone，可以选择比如iPad等其他用户组件，也可以进行画面的纵、横向切换等。

用鼠标拖拽组件

开发方式和以前的Delphi、Visual Basic一样，也是通过用鼠标拖拽组件来完成。从工具箱选择组件，然后拖拽到设计页面上。

比如，想要显示iPhone应用程序中常见的蓝色标题栏，只需把一个TToolBar组件拖拽到设计页面上即可（图3）。标题栏上表示的文字用TLabel进行定义。

可以对应画面旋转和不同的解像度

开发移动应用程序时，因为存在画面旋转等，所以需要把控好页面布局。Delphi可以利用几个布局控制功能，实现在不同解像度设备上的显示、画面旋转等。

比如，通过把配置在TToolBar上的TLabel的Align属性设为alContents，TLabel就可以占满整个标题栏（图4）。

利用组件进行编码

这次我们使用TTimer组件，试着做一个“倒

图1　新建移动应用程序

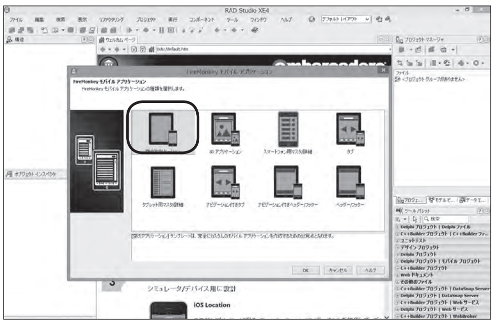


图2　图形化设计页面

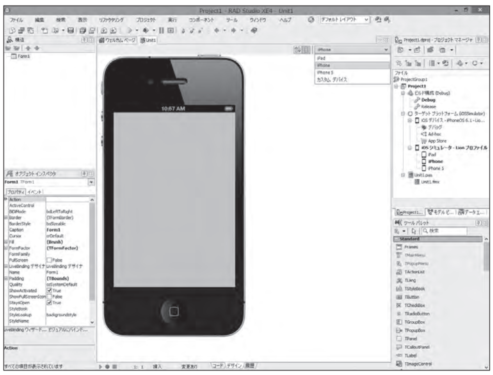


图3　TToolBar以及在其上配置的TLabel

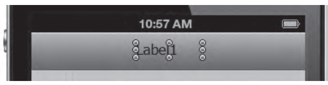


图4　占满整个TToolBar的TLabel



计时Hello World”。点击按钮，就会3、2、1进行倒计时，然后显示Hello World。

TTimer是一个组件，它可以提供以一定间隔执行处理的功能。这样，画面上显示的要素以外的逻辑，也进行了组件化，所以使用很方便。

OnTimer事件的代码如下所示 。

```
procedure TForm1.Timer1Timer(Sender:
TObject);
begin
    if Counter = 0 then
    begin
        Timer1.Enabled := False;
        Label2.Text := 'Hello World';
    end
    else
    begin
        Label2.Text := IntToStr(Counter);
        Dec(Counter);
    end;
end;
```

使用模拟装置或实体机进行调试

然后，可以在模拟装置或实体机上，运行完成的iOS应用程序。若使用实体机，需要USB连接。不过不管是哪种情况，运行都需要Mac环境。最近，也有的服务提供在云端通过远程连接运行的iOS环境，所以即便只有Windows机也可以进行开发了。

在模拟装置上运行的话，结果如图5所示。如果在代码中设置了断点，不管是模拟装置还是实体机，程序都会被中断，从而可以进行逐句执行和变量的监视等。

使用Delphi进行iOS开发的优势

虽然只是粗略地介绍了一下用Delphi进行

图5　执行应用程序（正在倒计时中）



的iOS开发，那么它与至今为止的开发有什么不同呢？又具有什么样的优势呢？

可以直观快速得到结果

移动开发的应用程序一般规模都较小，且工期都很短。如果使用Delphi，能够做成立即可以运行的应用程序，从而实现敏捷式开发。

组件丰富，可以实现分工合作

在试用Delphi时，还有一点很让人吃惊，那就是组件数非常多。UI自不必说，就是互联网通信、访问数据库、设备功能等都进行了组件化，所以只用很少的代码量，就可以实现很复杂的、高端的功能。

组件支持用户自定义。技术水平高的工程师可以自定义组件，年轻的经验少的工程师，可以利用这些组件来开发应用程序。这就实现了分工合作。

与Windows开发并存、移植更容易

Delphi原本是面向Windows的开发工具，现在Windows、Mac、iOS都可以支持，以后可能还会支持Android。拥有可以对应多个平台的组件框架，使得同样的代码既可以面向PC开发又可以面向移动设备开发。这样就可以减轻Windows应用程序向iPhone、iPad移植所花费的工夫，也有益于以Windows和移动设备双平台为目标的开发。

对Delphi的期待

可以对应iOS的Delphi版本已经有相关书籍出版了，今后也很期待相关技术信息的进一步充实。2013年后半年发布了支持Android的版本，同样的代码已可以对应多平台的设备。而且，计划2014年C++也可以完成这样的功能。很期待其今后的发展。

Delphi XE4的免费试用版可以从以下网址下载：

http://www.embarcadero.com/cn/products/delphi。

创造互联网服务未来的人们 25

探索CyberAgent公司的网络基础设施（后篇）

采访/撰文：Insight Image　川添贵生KAWAZOE Takao mail@insightimage.jp　　摄影：中村俊哉 NAKAMURA Toshiya
译/王凤波

对于支撑互联网服务的网络基础设施平台，今后有着怎样的发展要求呢？针对CyberAgent公司网络基础设施的发展前景，继上篇报道之后，我们又一次采访了篠原雅和、高桥哲平和村越俊克这三位工程师。

以实现基础设施构建自动化为目标的CyberAgent

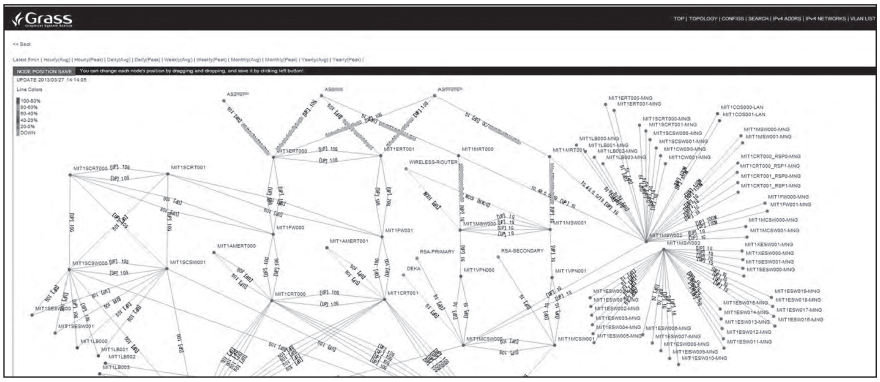
为了整合快速部署服务及应用的体制，目前DevOps正在迅速普及，这是一种通过开发工程师（Developer）和运维工程师（Operator）的密切合作，可以实现基础设施构建自动化的构想。例如，当需要发布一种新的服务时，以往的业务流程一般是由开发工程师首先提出申请，然后再由运维工程师构建基础设施，在此基础之上才能对外发布该服务。而DevOps旨在通过基础设施构建流程的自动化来实现快速的服务部署。

这种DevOps思想也渗透到了CyberAgent公司。今年春天曾经参与新服务基础设施网络部分设计与构建工作的哲平先生，对于今后的发展目标做了如下阐述。

“Web页面上执行简单的点击操作，就能自动生成包括负载均衡和防火墙等在内的网络环境。说得再极端些，即使客户要求从明天开始就着手某一服务的开发工作，我们也只需要进行简单的点击操作，便能轻松构建好服务器以及网络环境，并立刻投入到该服务的开发工作当中。希望实现这样的环境能够为公司的业绩作出贡献，这是我们的伟大使命之一。”（哲平）

另外，现阶段公司内部系统的防火墙构建等工作可以进行外包，委托外包方来为我们实现，从某种意义上而言，这也是一种形式的自动化。

图1　通过网络虚拟化实现数据中心的整合




照片1　村越（左）、哲平（中）、篠原（右）

但是篠原先生提出了更高的目标："我们最终要通过软件技术来实现自动化，并且完全摆脱人的手动操作。"

接下来，村越先生又从另一个侧面进一步为我们讲述了实现自动化的优势。

"例如，对于操作失误导致的故障，目前我们所采取的应对措施往往是改善设置的检查体制，或者对设置内容作记录等，而这种应对措施往往会再次引发同样的故障。但是，如果这种操作改为由系统来执行的话，我们就可以通过修改系统而防止同样故障再次发生，从而也就提高了我们服务的质量。"（村越）

另外，CyberAgent正计划通过网络虚拟化技术来整合现有的多个数据中心。

"现在4个数据中心都是各自独立运营的，上层领导们希望通过SDN（Software Defined Network）等技术将分散的数据中心虚拟到一起，从而可以确保在使用过程中不用在意其物理位置的分离。虽然现在还没有这样的业务需求，但是将来如果某个服务的开发项目需要考虑灾后重建的时候，像这样通过虚拟技术将数据中心整合到一起，会比现有的基础设施更容易实现。"（哲平）

"另外还有一个可以预见的优势，那就是可以防止服务或者基础设施被锁定在某个特定的数据中心之内。具体而言，当运行在某个数据中心的服务需要迁移到其他数据中心时，往往会受到基础设施方面的制约而难以实现。但是，像这样将基础设施进行扁平化之后，这种迁移就会变得非常容易实现。"（篠原）

哲平先生用"SKBNW"这一关键字向我们了讲述了未来网络环境的理念。

"容易修正（S）、构建（K）简单、可分离（B）型网络（NW），就是所谓的SKBNW（笑）。实现这种高度灵活的网络将是我们今后追求的目标。"（哲平）

"现阶段，我和高桥正在思考如何才能实现这一目标，以及需要什么样的技术。目前我们想到的最好的方法是利用OpenFlow来实现网络的虚拟化。当然，现有的协议或许也能够完成类似的操作，但是那样的话，很有可能最终使用的还是各制造商独自拥有的功能。虽然OpenFlow还存在着一些有待完善的地方，但是我想我们努力的这个方向是正确的。"（篠原）

值得我们关注的是，在未来网络环境的理念当中融入了"容易修正"的概念。对于曾经含辛茹苦构建地网络环境我们不能一味地坚守，还要积极顺应时势的变迁，勇于创造新的价值。我想，具有这种敢于舍弃、勇于创新的精神正是CyberAgent如此强大的原因之一吧。

看完了

如果您对本书内容有疑问，可发邮件至contact@turingbook.com，会有编辑或作译者协助答疑。也可访问图灵社区，参与本书讨论。

如果是有关电子书的建议或问题，请联系专用客服邮箱：ebook@turingbook.com。

在这里可以找到我们：

微博 @图灵教育 : 好书、活动每日播报
微博 @图灵社区 : 电子书和好文章的消息
微博 @图灵新知 : 图灵教育的科普小组
微信 图灵访谈 : ituring_interview，讲述码农精彩人生
微信 图灵教育 : turingbooks